Autoencoders and Autoregressive Models

Deep Learning (DSE316/616)

Vinod K Kurmi
Assistant Professor, DSE

Indian Institute of Science Education and Research Bhopal

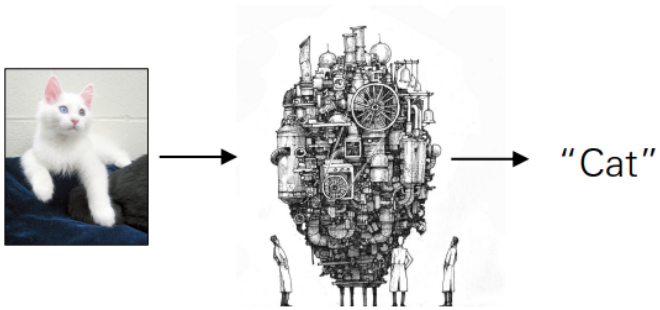
Oct 13, 2022

Disclaimer

- Much of the material and slides for this lecture were borrowed from
  - Bernhard Schölkopf's MLSS 2017 lecture,
  - Tommi Jaakkola's 6.867 class,
  - CMP784: Deep Learning Fall 2021 Erkut Erdem Hacettepe University
  - Fei-Fei Li, Andrej Karpathy and Justin Johnson's CS231n class
  - Hongsheng Li's ELEG5491 class
  - Tsz-Chiu Au slides
  - Mitesh Khapra Class notes

Discriminative vs. Generative Models

$$p(y|x)$$

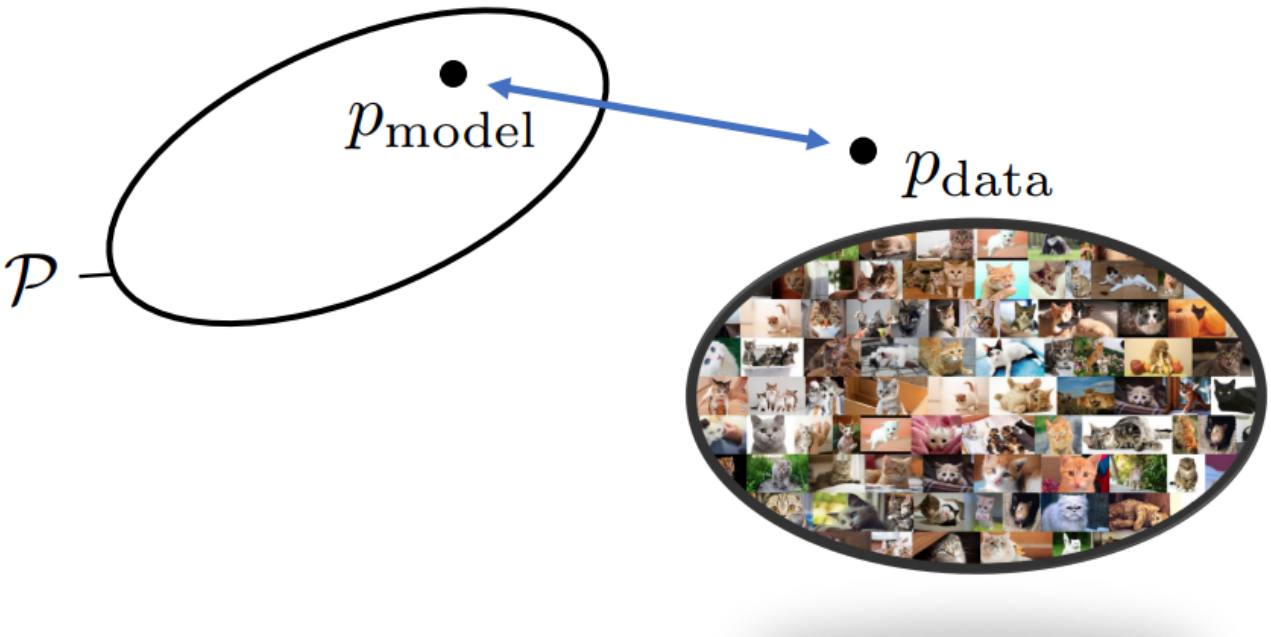


Discriminative models

$$p(x|y)$$

Generative models

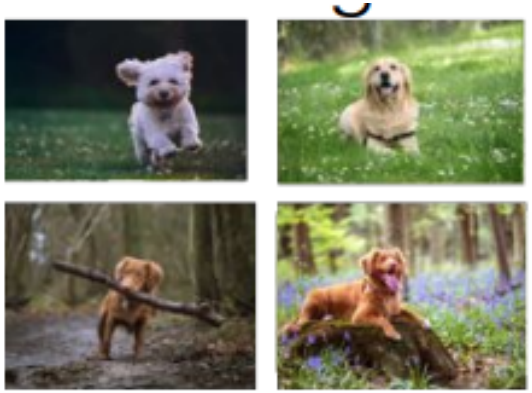
Generative Modeling



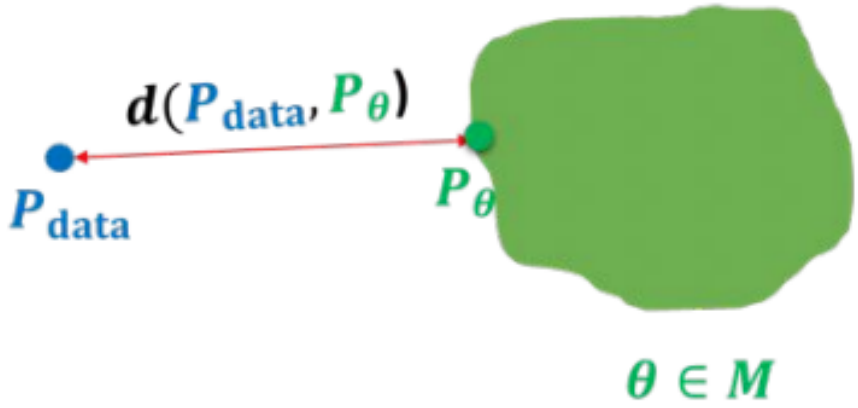
- **Goal:** Learn some underlying hidden structure of the training samples to generate novel samples from same data distribution

Learning a generative model

- We are given a training set of examples, e.g., images of dogs

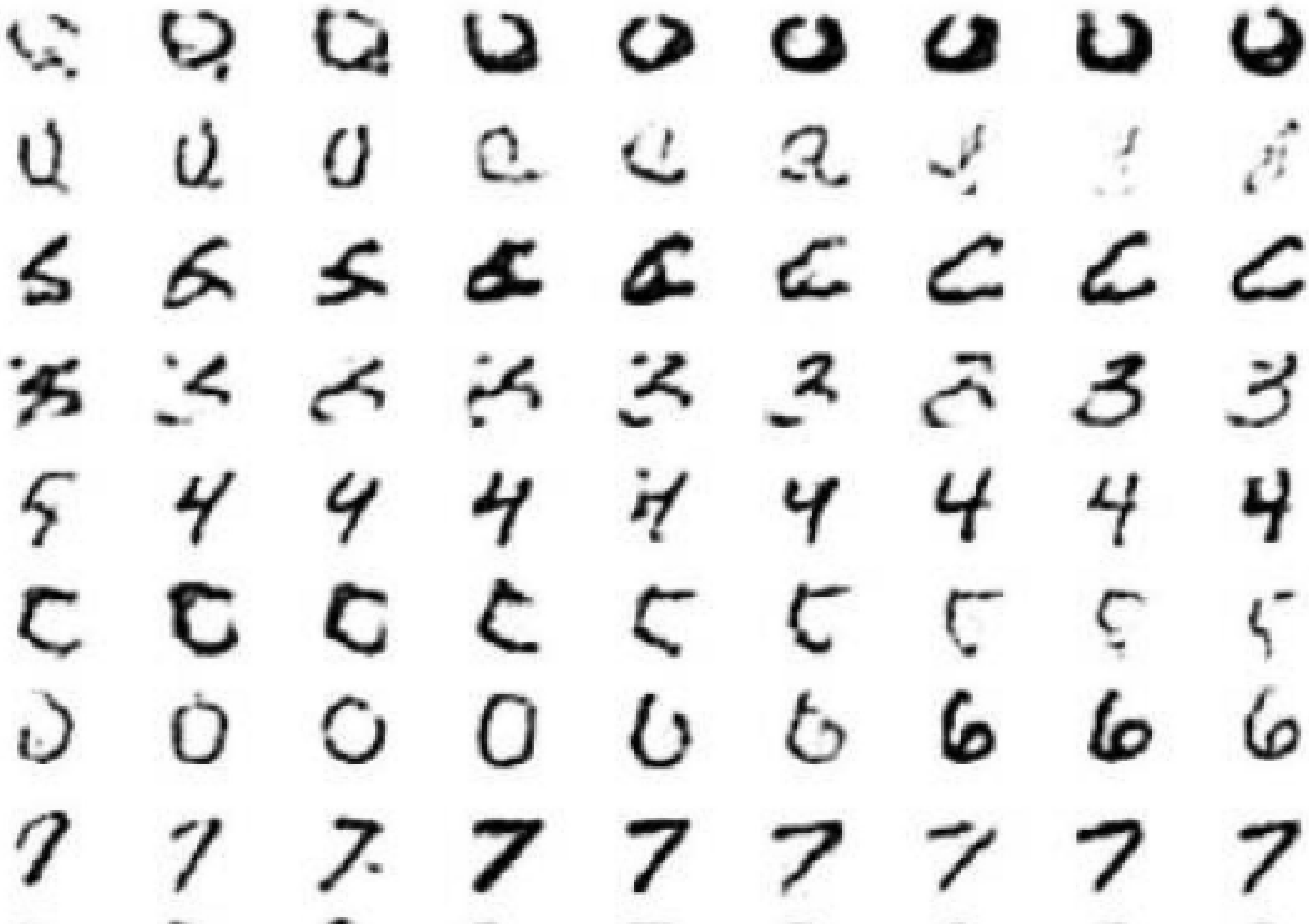


$$x_i \sim P_{\text{data}}$$
$$i = 1, 2, \ldots, n$$

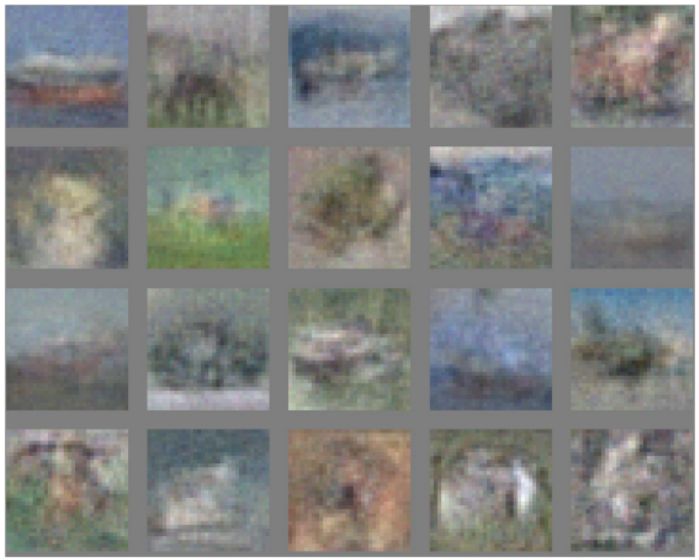


- We want to learn a probability distribution $p(x)$ over images x s.t.
  - **Generation:** If we sample x_{new} $p(x)$, x_{new} should look like a dog (sampling)
  - **Density estimation:** $p(x)$ should be high if x looks like a dog, and low otherwise (anomaly detection)
  - **Unsupervised representation learning:** We should be able to learn what these images have in common, e.g., ears, tail, etc. (features)

Generate Images

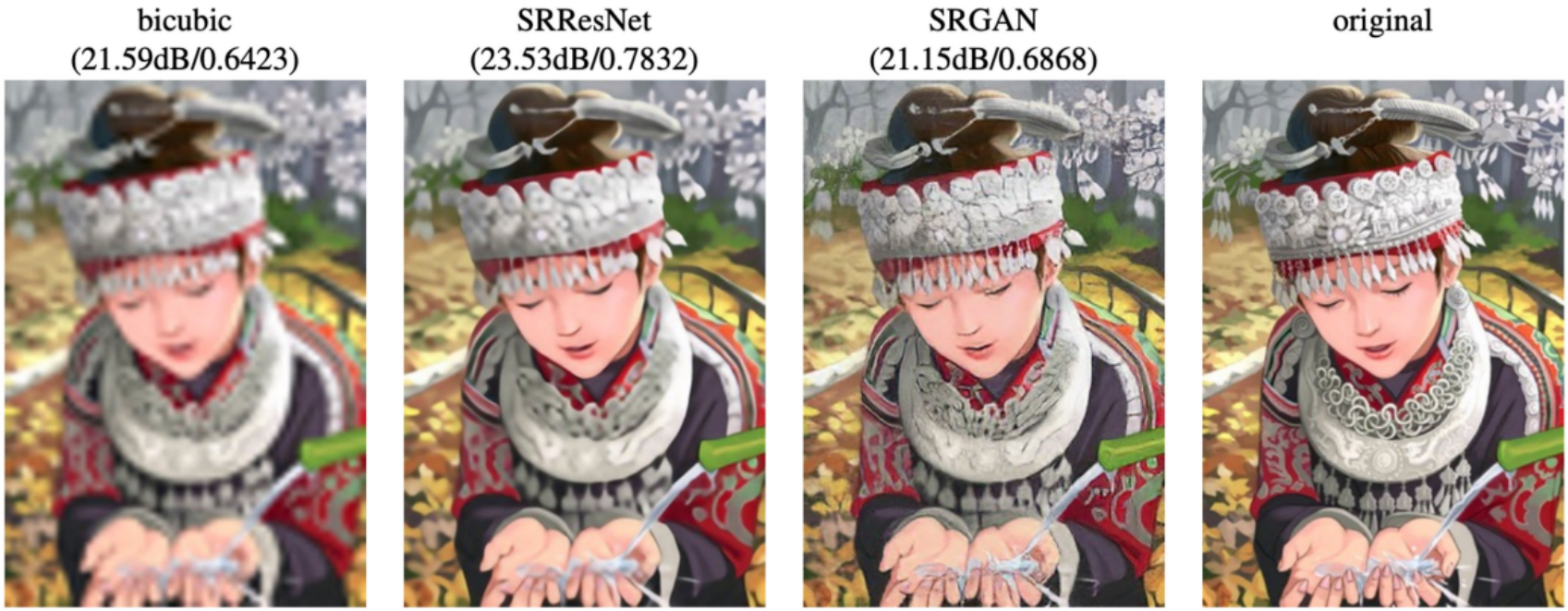


Generate Images



Generate Images

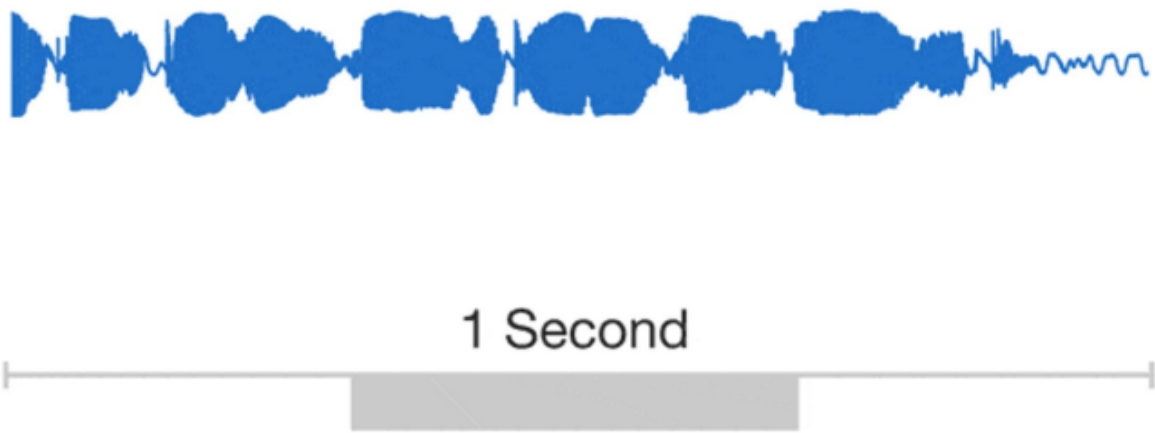
Generate Images



Generate Images

Generate Images

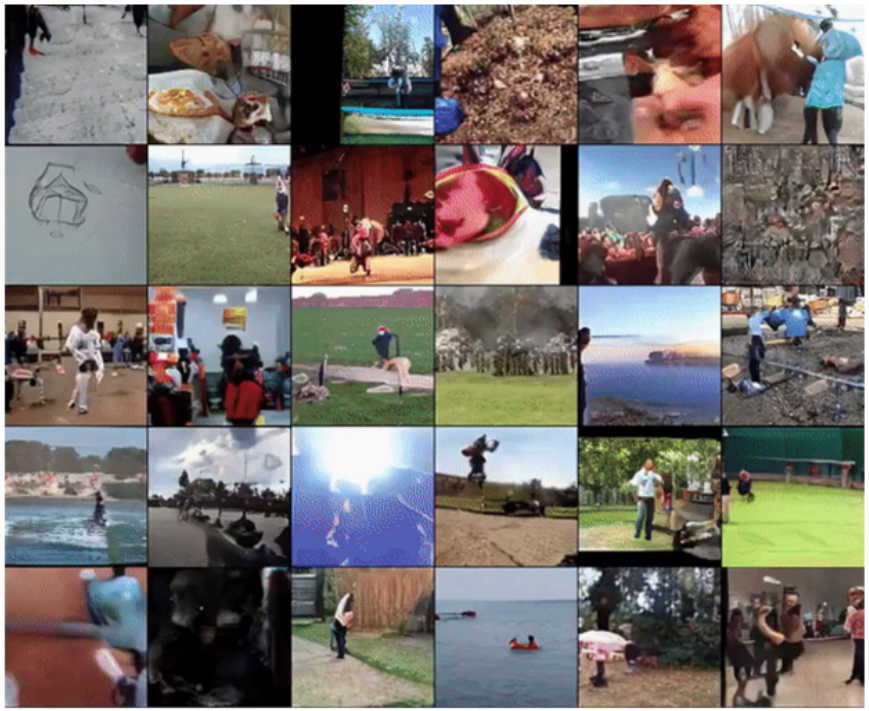
Generate Audio



Parametric

WaveNet

Generate Video



DVD-GAN: Adversarial Video Generation on Complex Datasets, Clark, Donahue, Simonyan, 2019

Generate Text

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.
```

[Char-rnn, karpathy, 2015]

Generate Math

```
\begin{proof}
We may assume that $\mathcal{I}$ is an abelian sheaf on
$\mathcal{C}$.
\item Given a morphism $\Delta : \mathcal{F} \to \mathcal{I}$
is an injective and let $\mathfrak q$ be an abelian sheaf on
$X$.
Let $\mathcal{F}$ be a fibered complex. Let $\mathcal{F}$ be a
category.
\begin{enumerate}
\item \hyperref[setain-construction-phantom]{Lemma}
\label{lemma-characterize-quasi-finite}
Let $\mathcal{F}$ be an abelian quasi-coherent sheaf on
$\mathcal{C}$.
Let $\mathcal{F}$ be a coherent $\mathcal{O}_X$-module. Then
$\mathcal{F}$ is an abelian catenary over $\mathcal{C}$.
\item The following are equivalent
\begin{enumerate}
\item $\mathcal{F}$ is an $\mathcal{O}_X$-module.
\end{lemma}
```

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the compariconly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

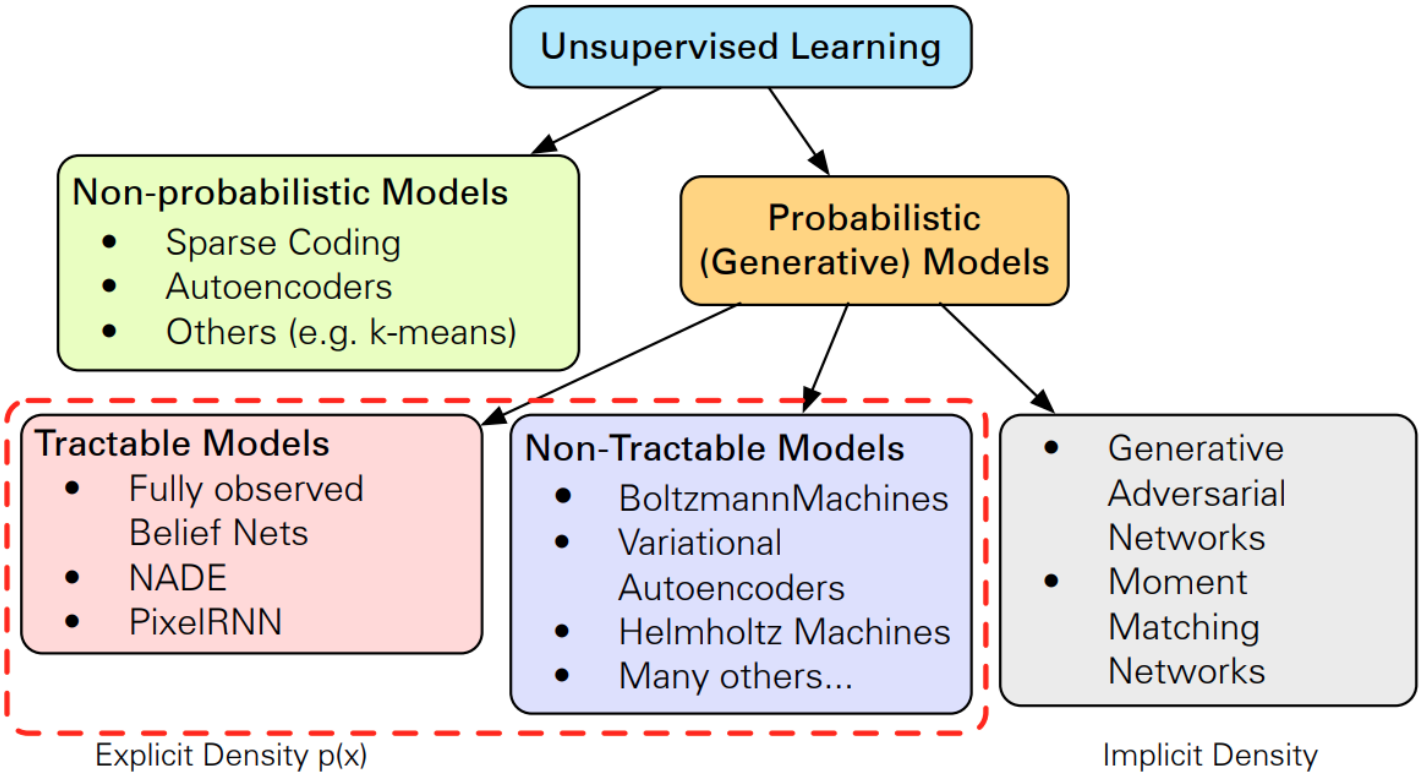
Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

[Char-rnn, karpathy, 2015]

Why Unsupervised Learning?

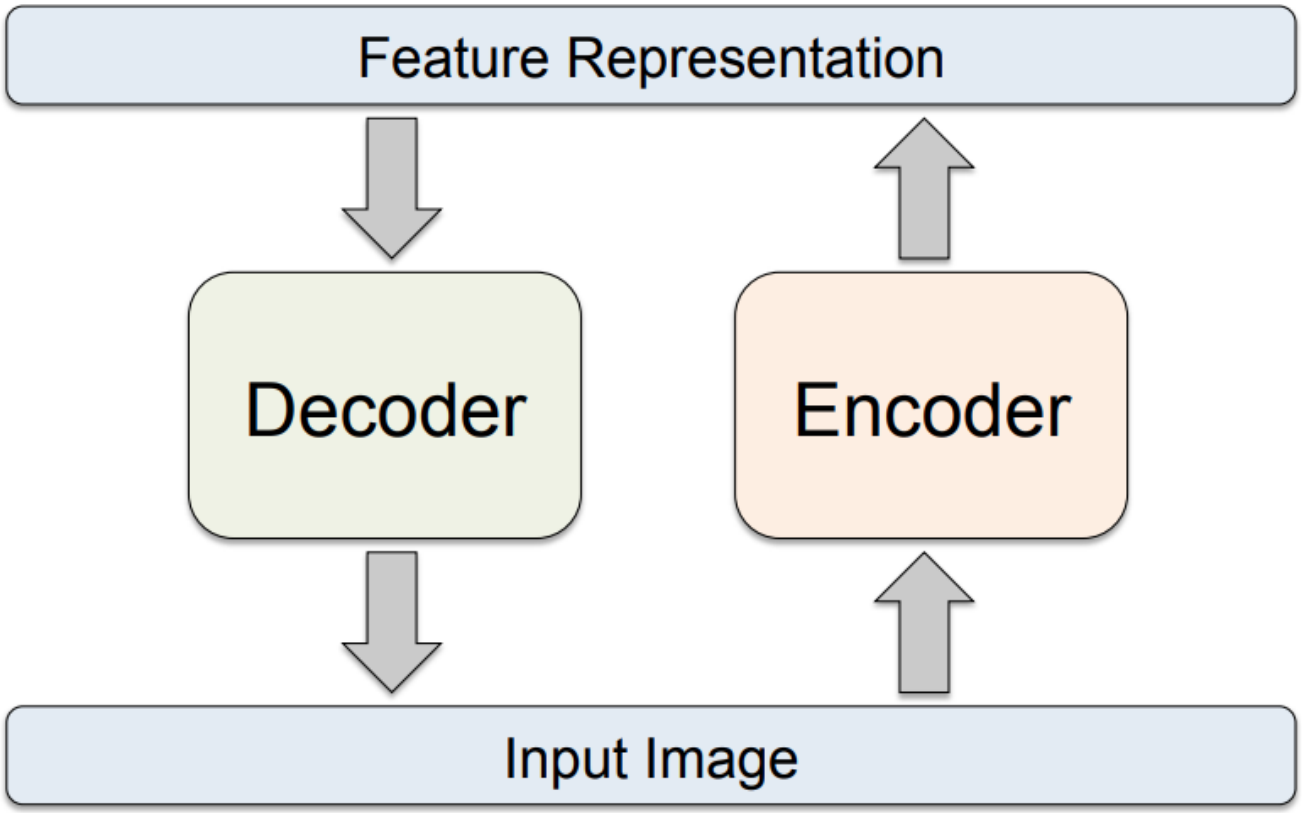
- Given high-dimensional data , we want to find a low-dimensional model characterizing the population.
- Recent progress mostly in supervised DL
- Real challenges for unsupervised DL
- Potential benefits:
- Exploit tons of unlabeled data
  - Answer new questions about the variables observed
  - Regularizer – transfer learning – domain adaptation
  - Easier optimization (divide and conquer)
  - Joint (structured) outputs


Unsupervised Learning
Non-probabilistic Models
• Sparse Coding
• Autoencoders
• Others (e.g. k-means)
Probabilistic (Generative) Models
Tractable Models
• Fully observed Belief Nets
• NADE
• PixelRNN
Non-Tractable Models
• BoltzmannMachines
• Variational Autoencoders
• Helmholtz Machines
• Many others...
• Generative Adversarial Networks
• Moment Matching Networks
Explicit Density p(x)
Implicit Density

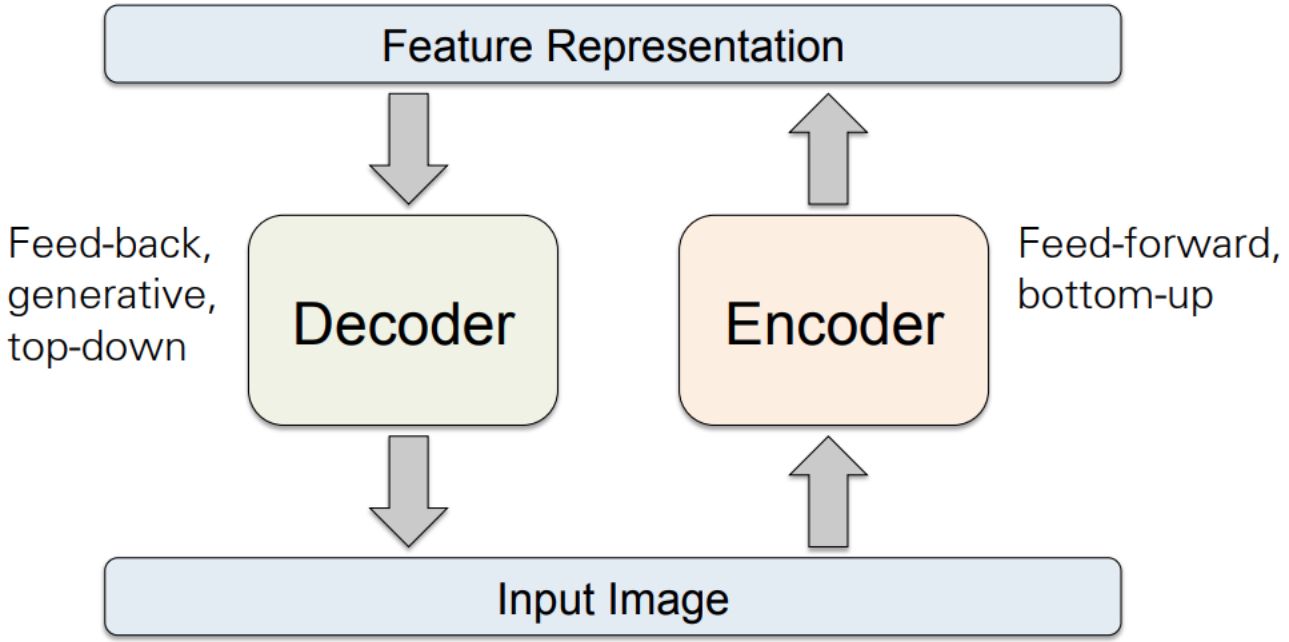
Basic Building Blocks:

- Autoencoders
- Variational Autoencoders
- Generative Adversarial Networks
- Normalizing Flow Models
- Diffusion models

Autoencoder

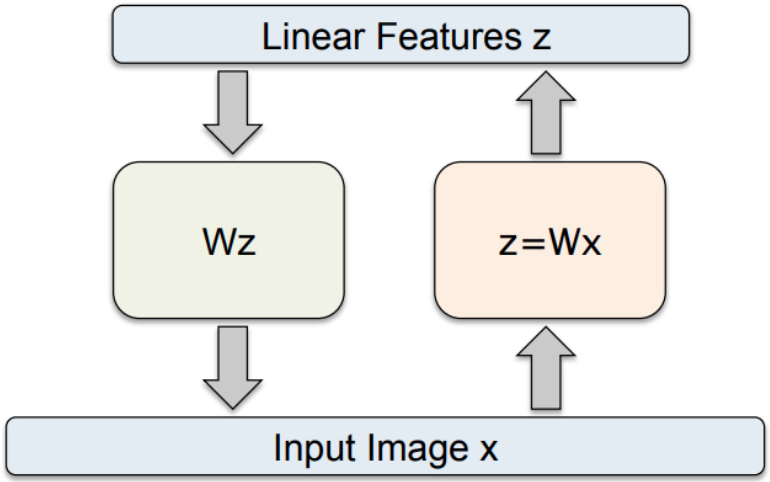


Autoencoder



- Details of what goes insider the encoder and decoder matter!
- Need constraints to avoid learning an identity

Autoencoder



- If the **hidden and output layers are linear**, it will learn hidden units that are a linear function of the data and minimize the squared error.
- The K hidden units will span the same space as the first k principal components. The weight vectors may not be orthogonal.

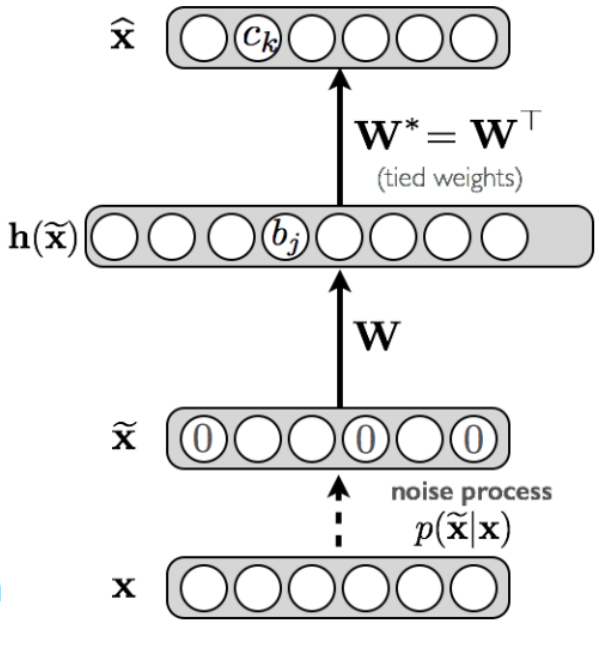
- With nonlinear hidden units, we have a nonlinear generalization of PCA.

Denoising Autoencoder

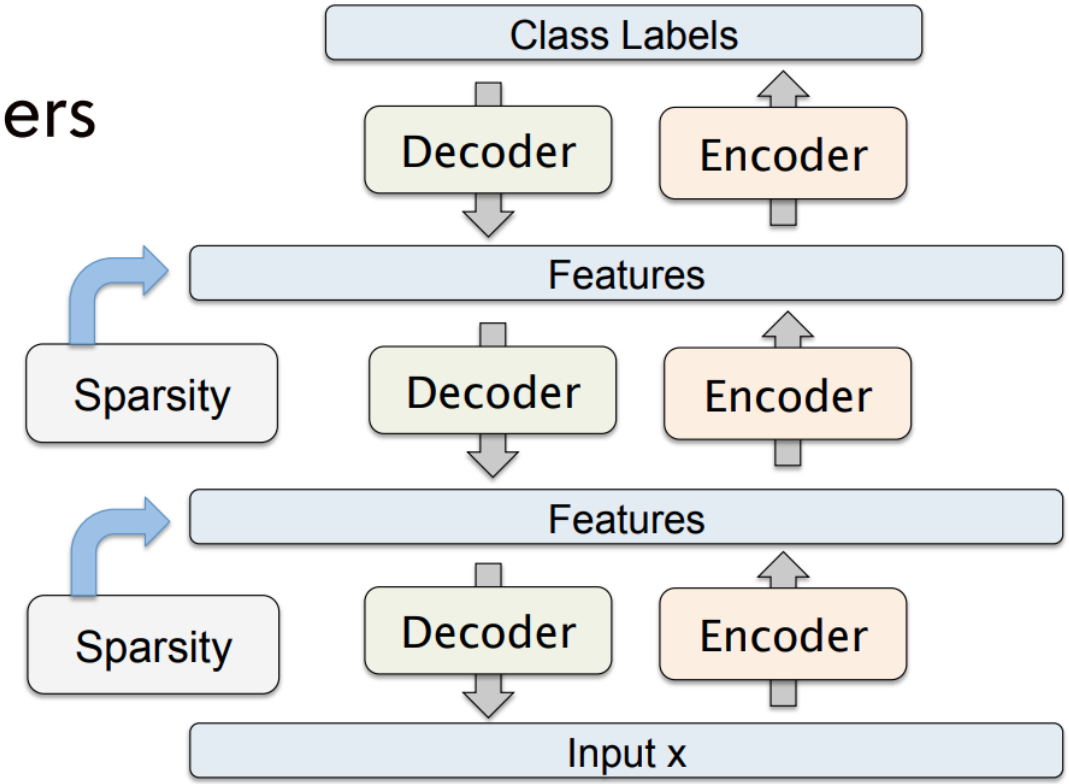
- **Idea:** Representation should be robust to introduction of noise:
  - random assignment of subset of inputs to 0, with probability ν
  - Similar to dropouts on the input layer
  - Gaussian additive noise

- **Reconstruction** $\widehat{\mathbf{x}}$ computed from the corrupted input $\widetilde{\mathbf{x}}$

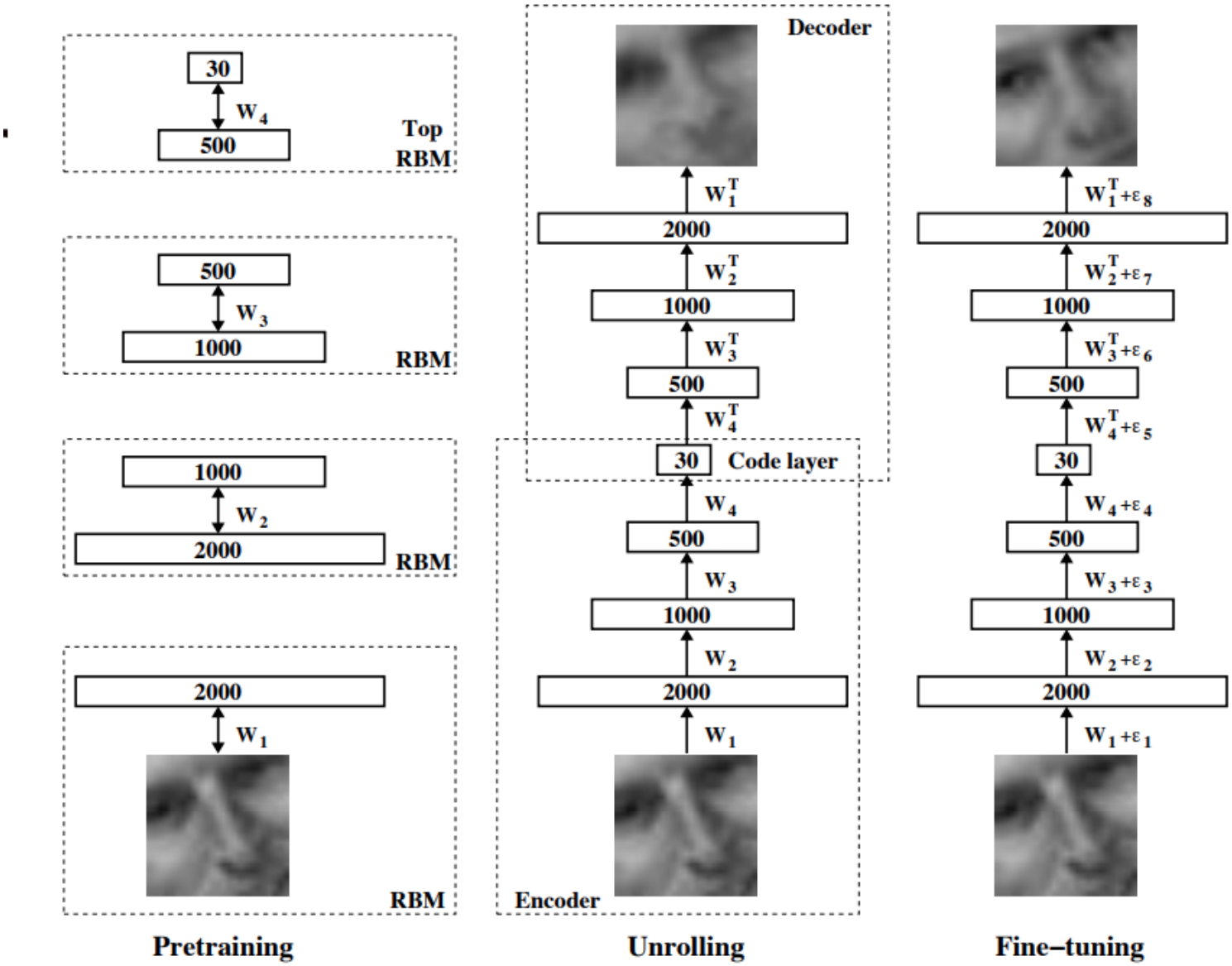
- **Loss function** compares $\widehat{\mathbf{x}}$ reconstruction with the noiseless input $\mathbf{x}$



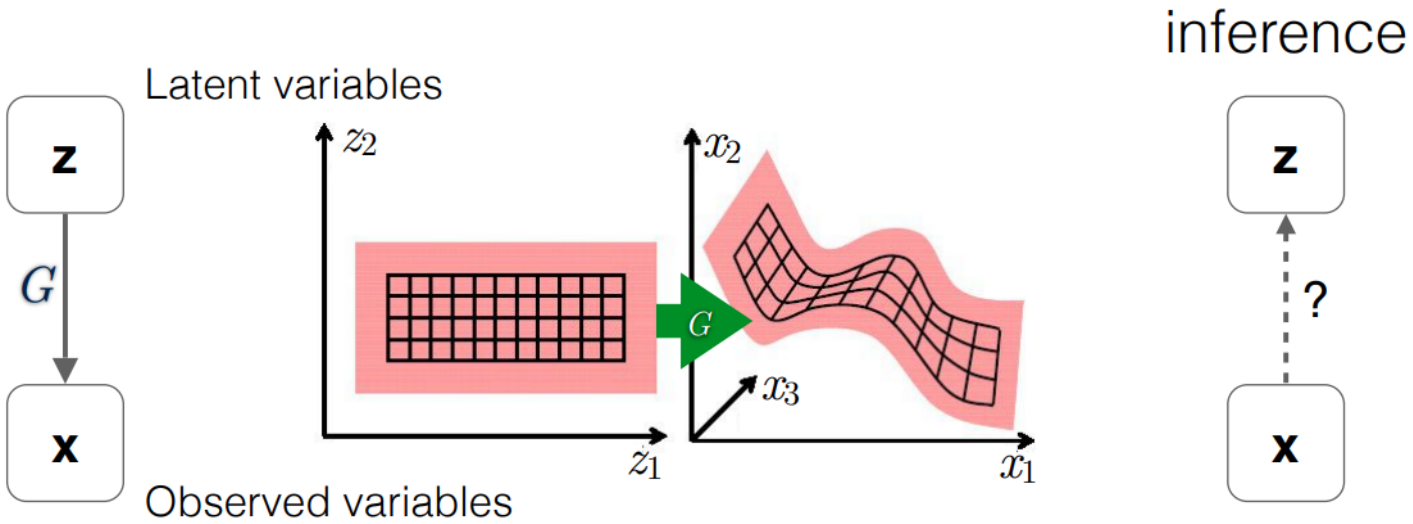
Stacked Autoencoders



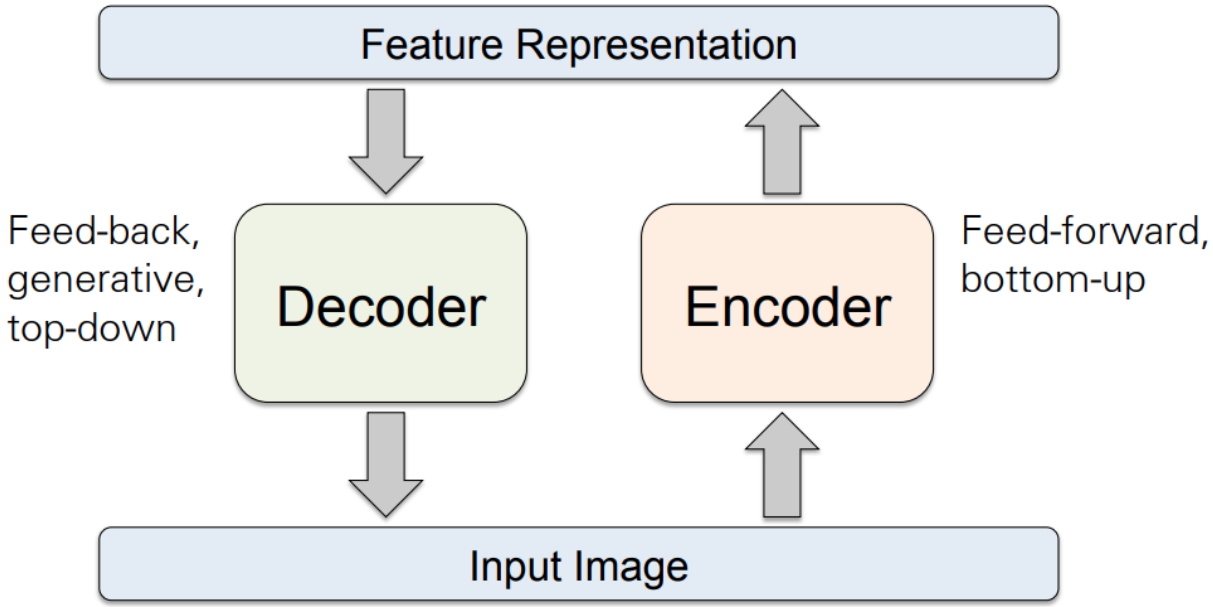
Deep Autoencoders



Latent variable model



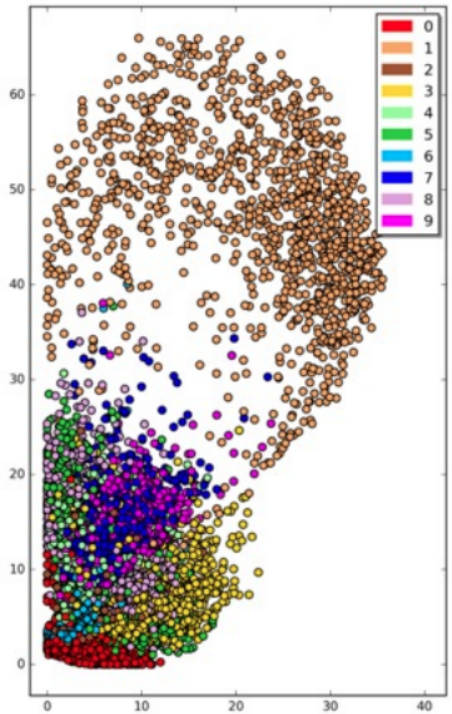
Autoencoders



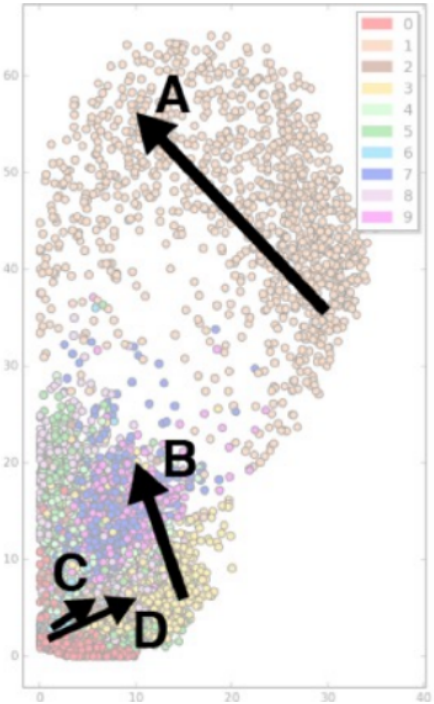
- Details of what goes insider the encoder and decoder matter!
- Need constraints to avoid learning an identity.

Parameter space of autoencoder

- Let's examine the latent space of an AE.
- Is there any separation of the different classes? If the AE learned the "essence" of the MNIST images, similar images should be close to each other.
- Plot the latent space and examine the separation.
- Here we plot the 2 PCA components of the latent space.



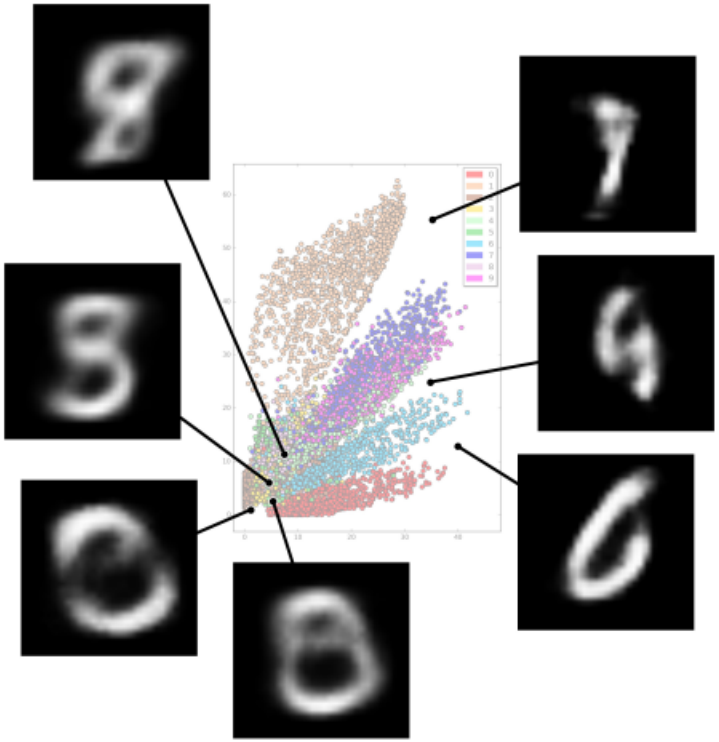
Traversing the latent space



- We start at the start of the arrows in latent space and then move to end of the arrow in 7 steps.
- For each value of z we use the already trained decoder to produce an image.

Problems with Autoencoders

- Gaps in the latent space
- Discrete latent space
- Separability in the latent space



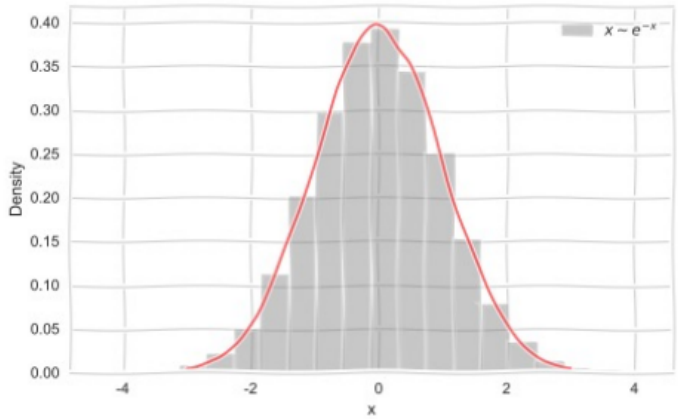
Generative models

- Imagine we want to generate data from a distribution,

$$x \backsim p(x)$$

- e.g.

$$x \backsim N(\mu, \sigma)$$



Generative models

- In other words we can think that if we choose *zbacksimUniform* then there is a mapping:

$$x = f(z)$$

such as

$$x \backsim p(x)$$

- where in general f is some complicated function.

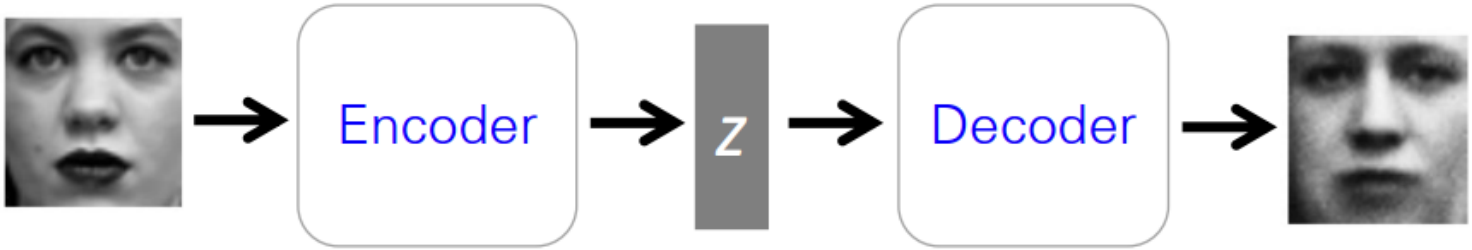
$$x \backsim N(\mu, \sigma)$$

- We already know that Neural Networks are great in learning complex functions

$$\boxed{z \sim g(z)} \longrightarrow \boxed{x = f(z)} \longrightarrow \boxed{x \sim p(x)}$$

Traditional Autoencoders

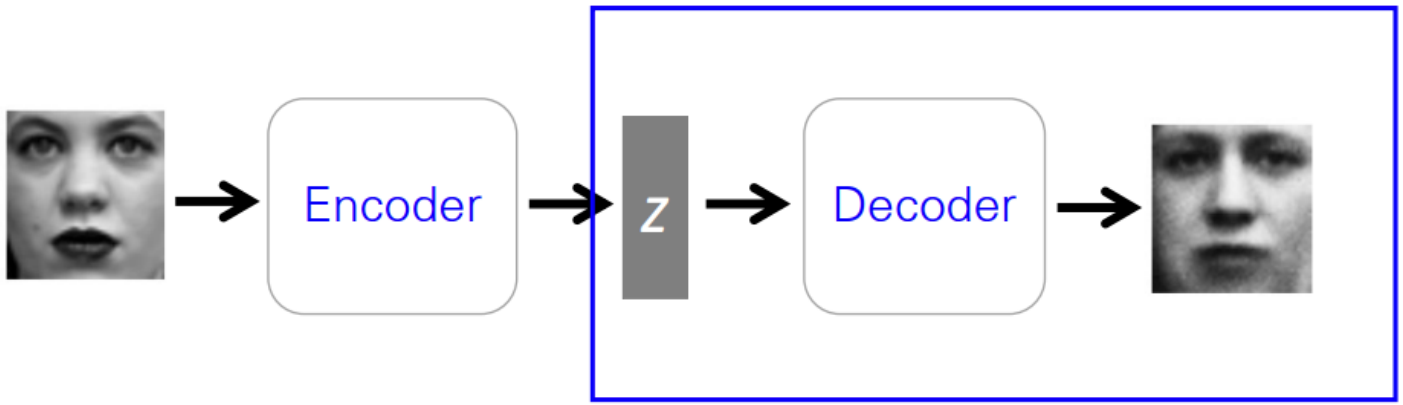
- In traditional autoencoders, we can think of encoder and decoders as some function mapping



$$z = h(x) \qquad \hat{x} = f(z)$$

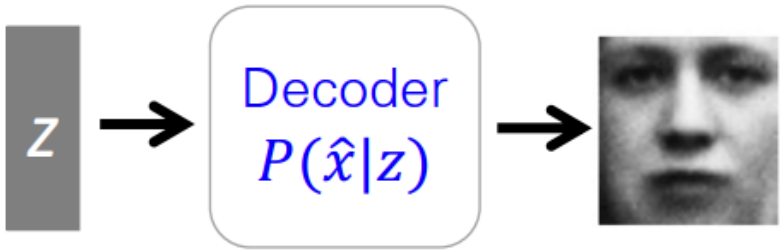
Variational Autoencoders

- To go to variational autoencoders, we need to first add some stochasticity and think of it as a probabilistic modeling.



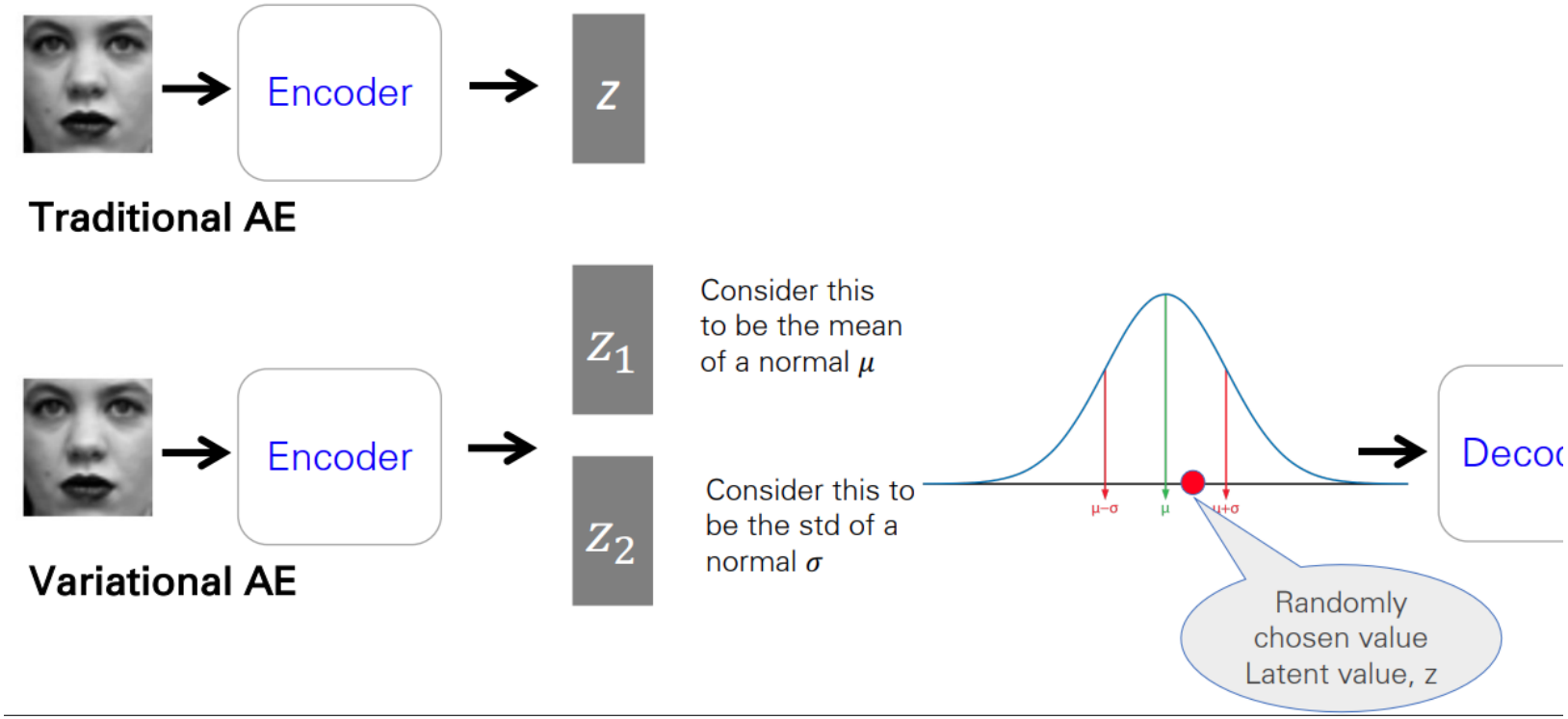
Variational Autoencoders

Sample from g(z)
e.g. Standard
Gaussian

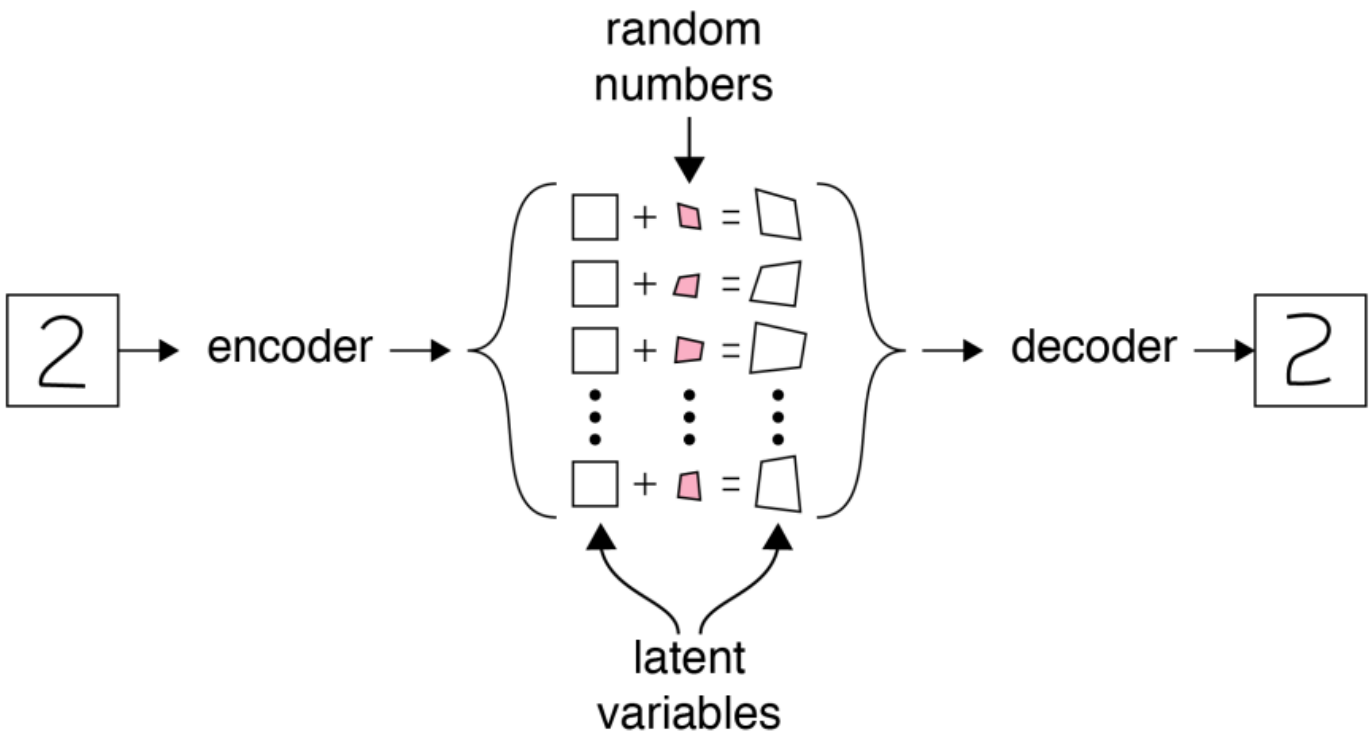


$$z \sim g(z) \qquad \hat{x} = f(z) \qquad \hat{x} \sim P(x|z)$$

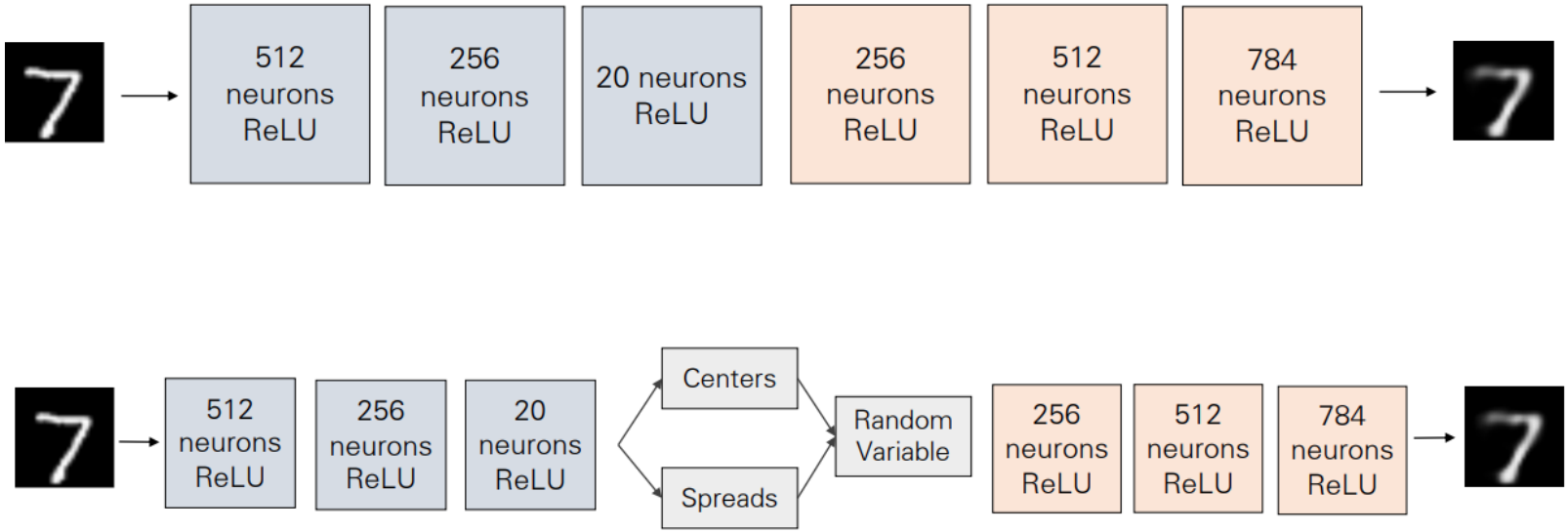
Variational Autoencoders



Variational Autoencoders

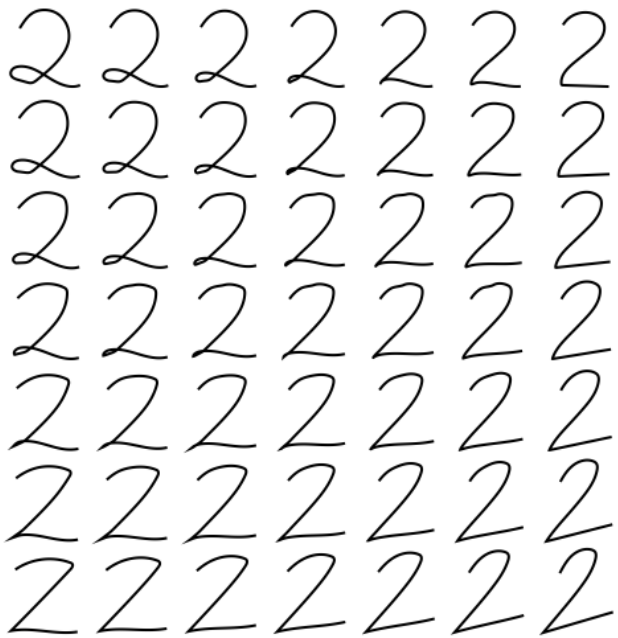


Variational Autoencoders

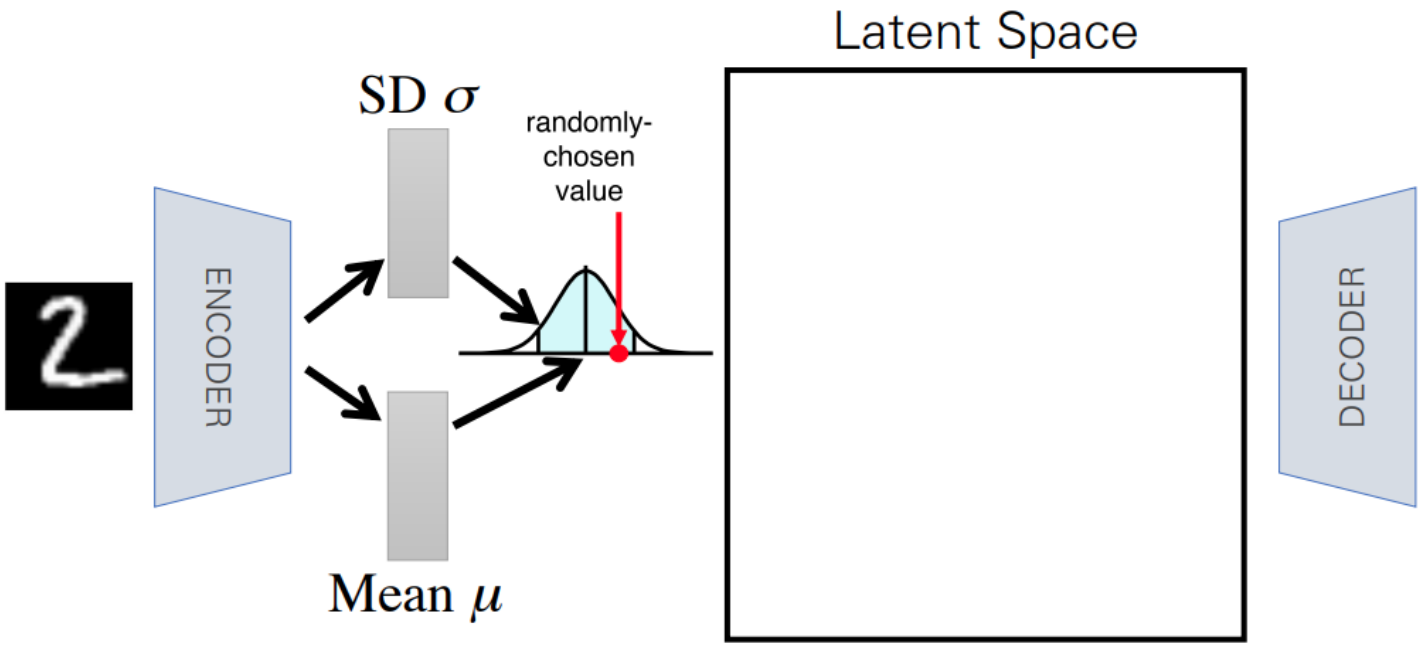


Separability in Variational Autoencoders

- Separability is not only between classes but we also want similar items in the same class to be near each other.
- For example, there are different ways of writing "2", we want similar styles to end up near each other.
- Let's examine VAE, there is something magic happening once we add stochasticity in the latent space.

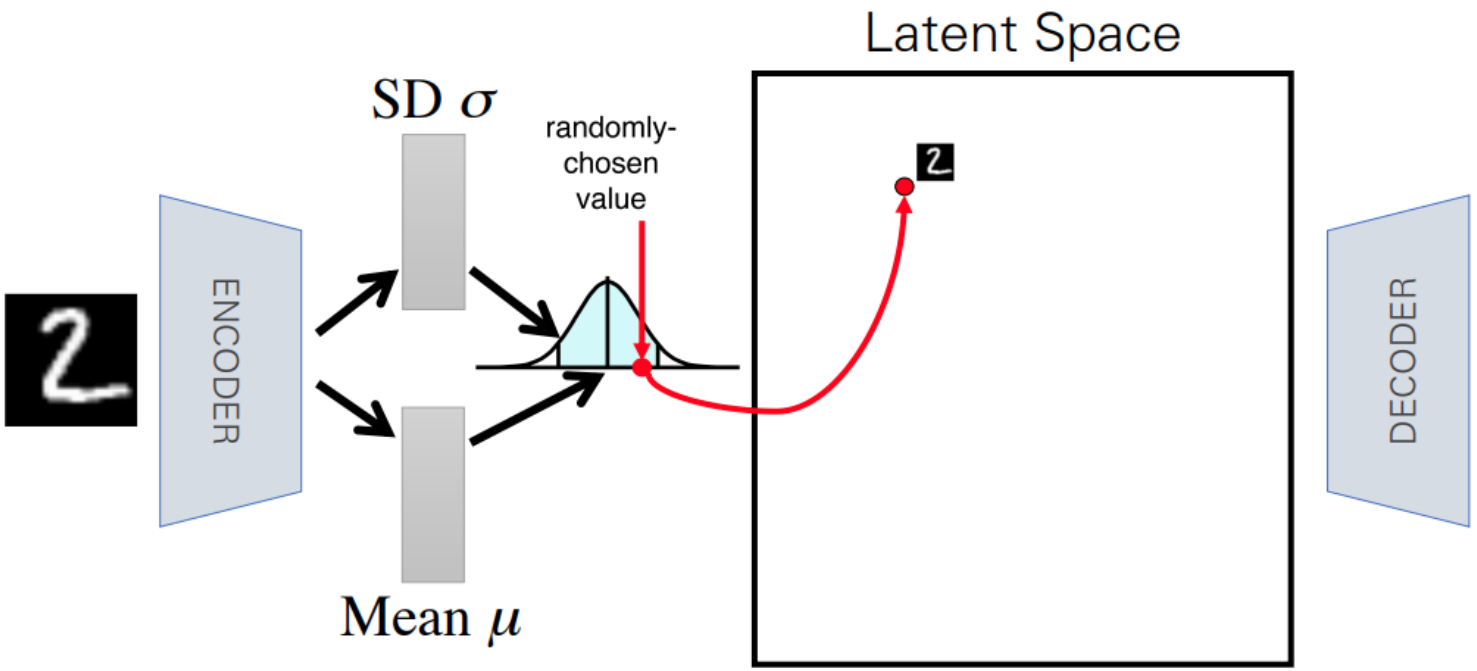


Separability in Variational Autoencoders



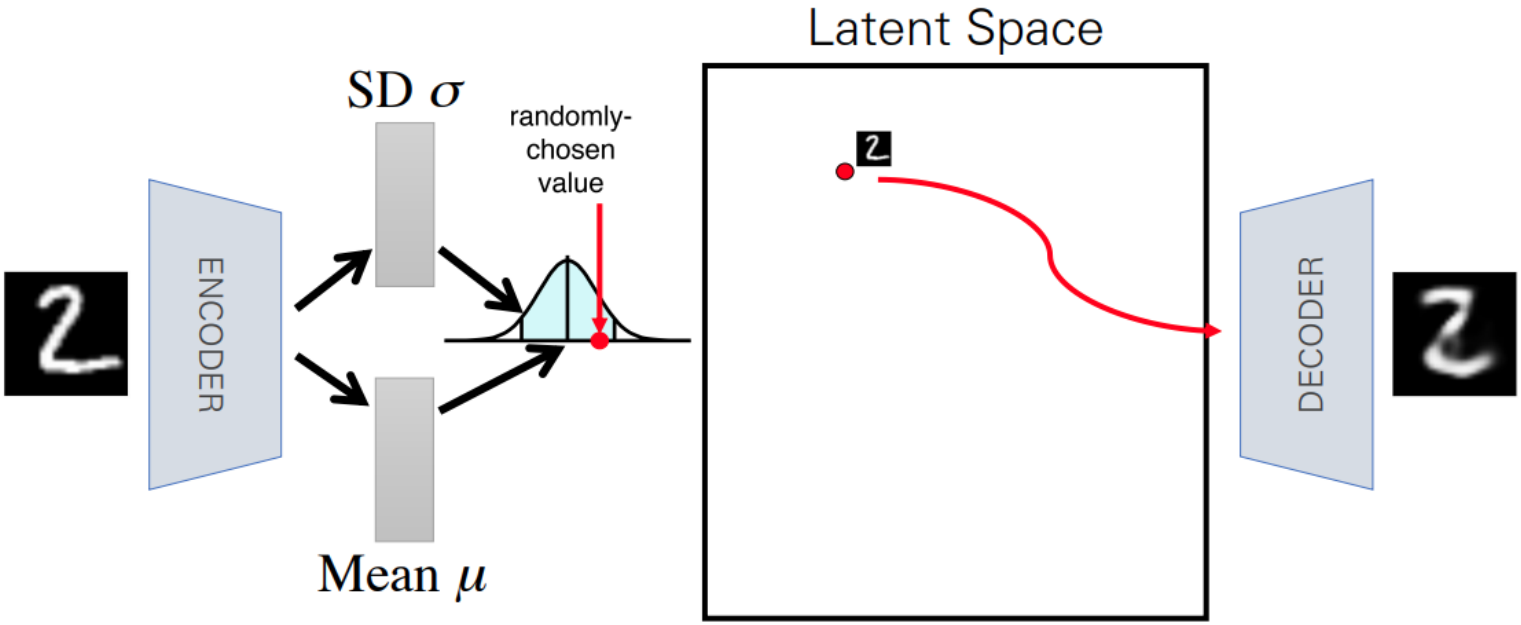
Encode the first sample (a "2") and find μ_1, σ_1

Separability in Variational Autoencoders



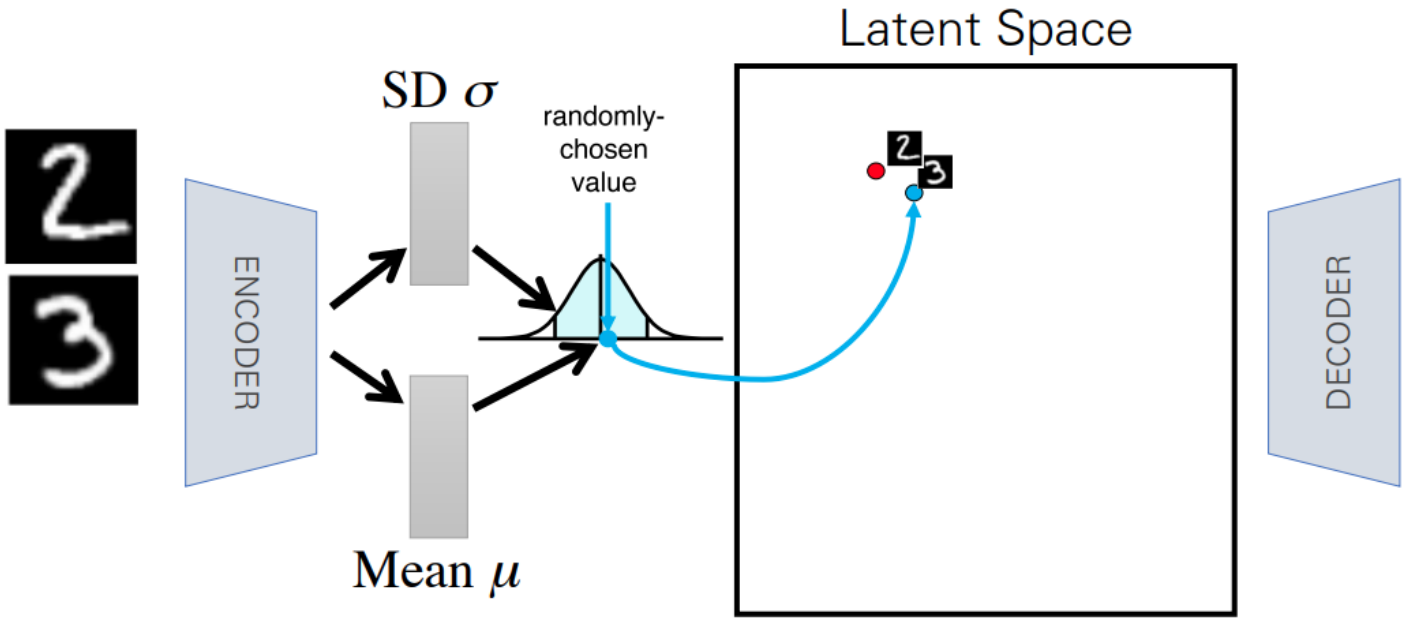
Sample $z_1 \sim N(\mu_1, \sigma_1)$

Separability in Variational Autoencoders



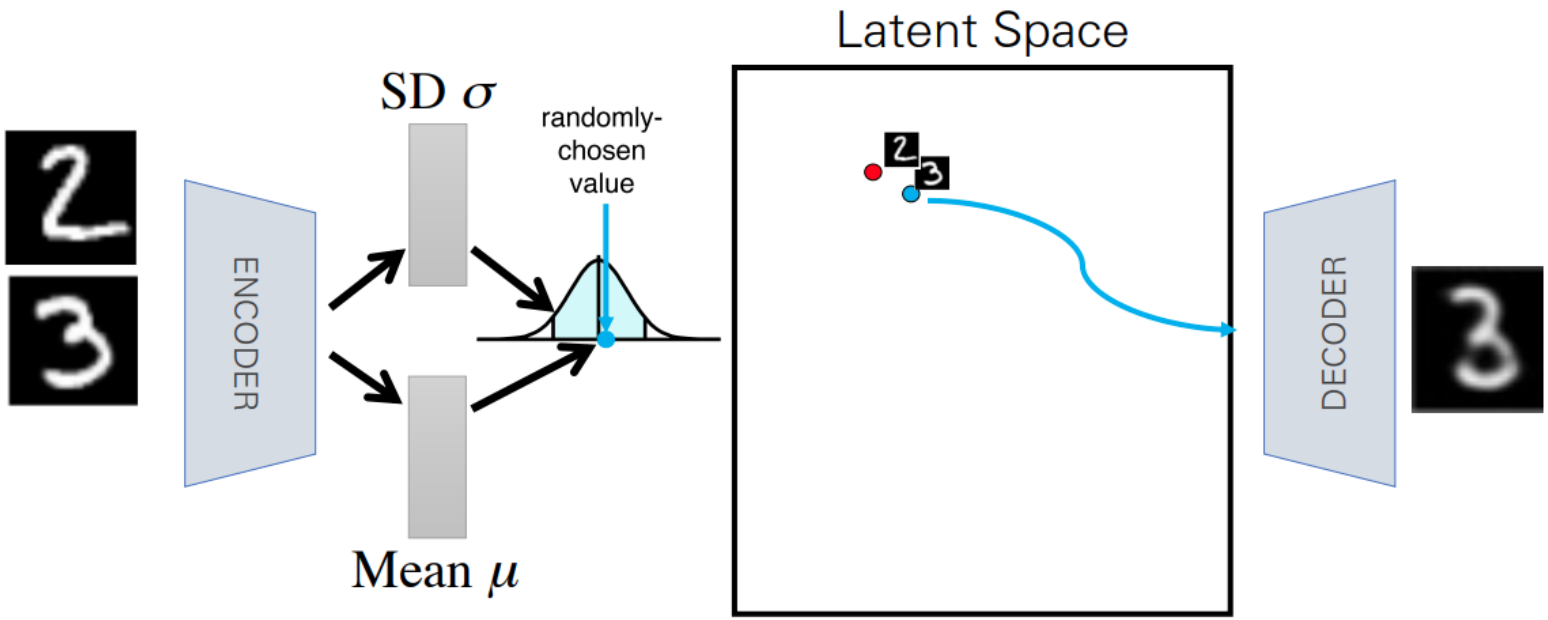
Decode to $\hat{x}_1$

Separability in Variational Autoencoders



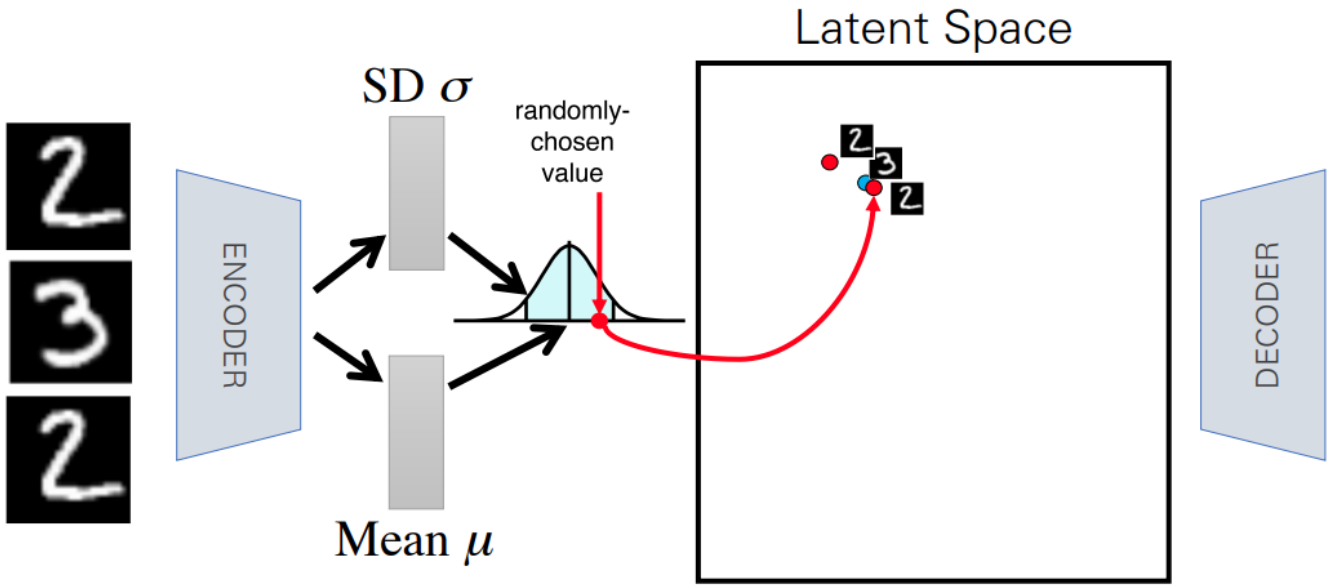
Encode the second sample (a "3") find μ_2, σ_2. Sample $z_2 \sim N(\mu_2, \sigma_2)$

Separability in Variational Autoencoders



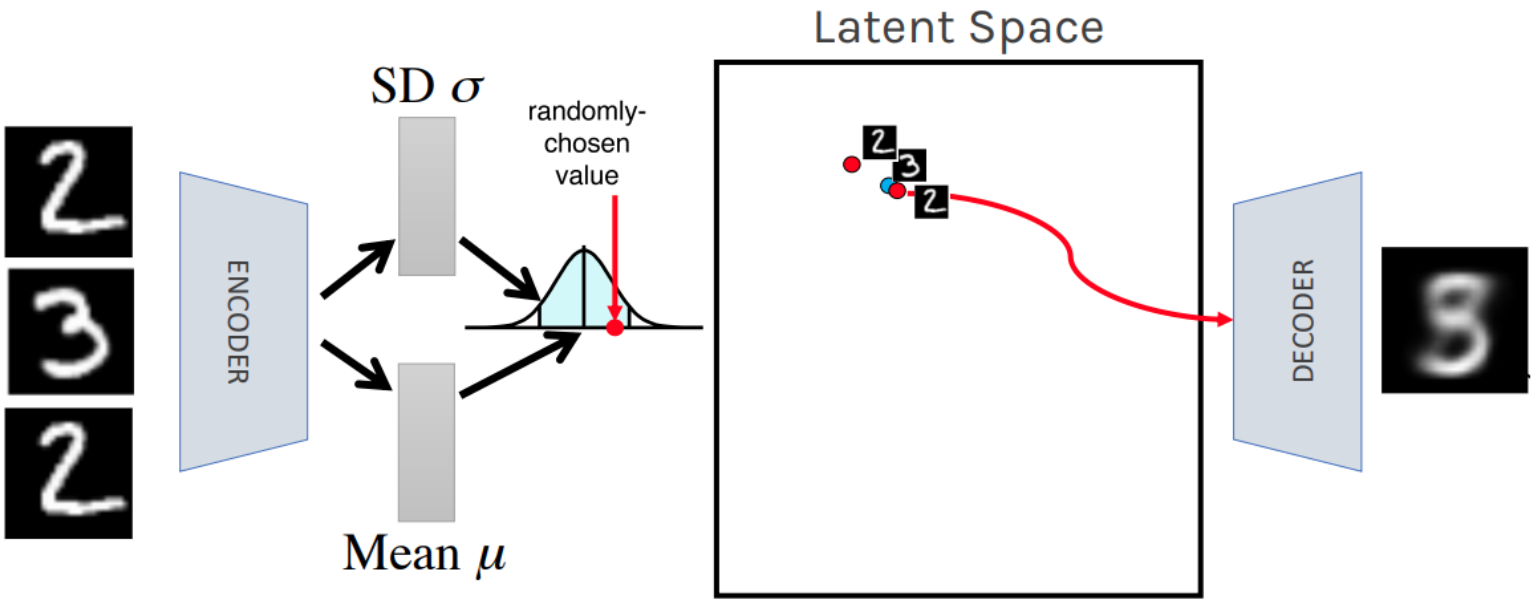
Decode to $\hat{x}_2$

Separability in Variational Autoencoders



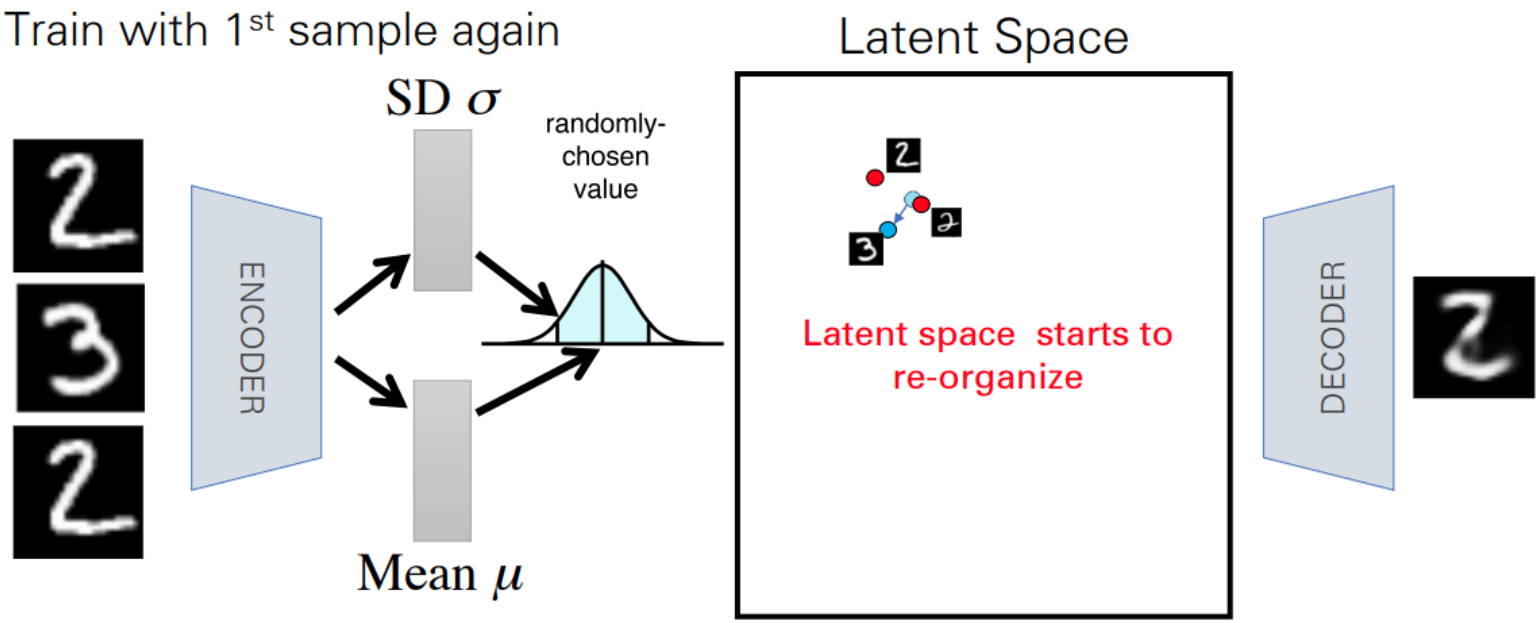
Train with the first sample (a "2") again and find μ_1, σ_1. However $z_1 \sim N(\mu_1, \sigma_1)$ **will not be the same**. It can happen to be close to the "3" in latent space.

Separability in Variational Autoencoders

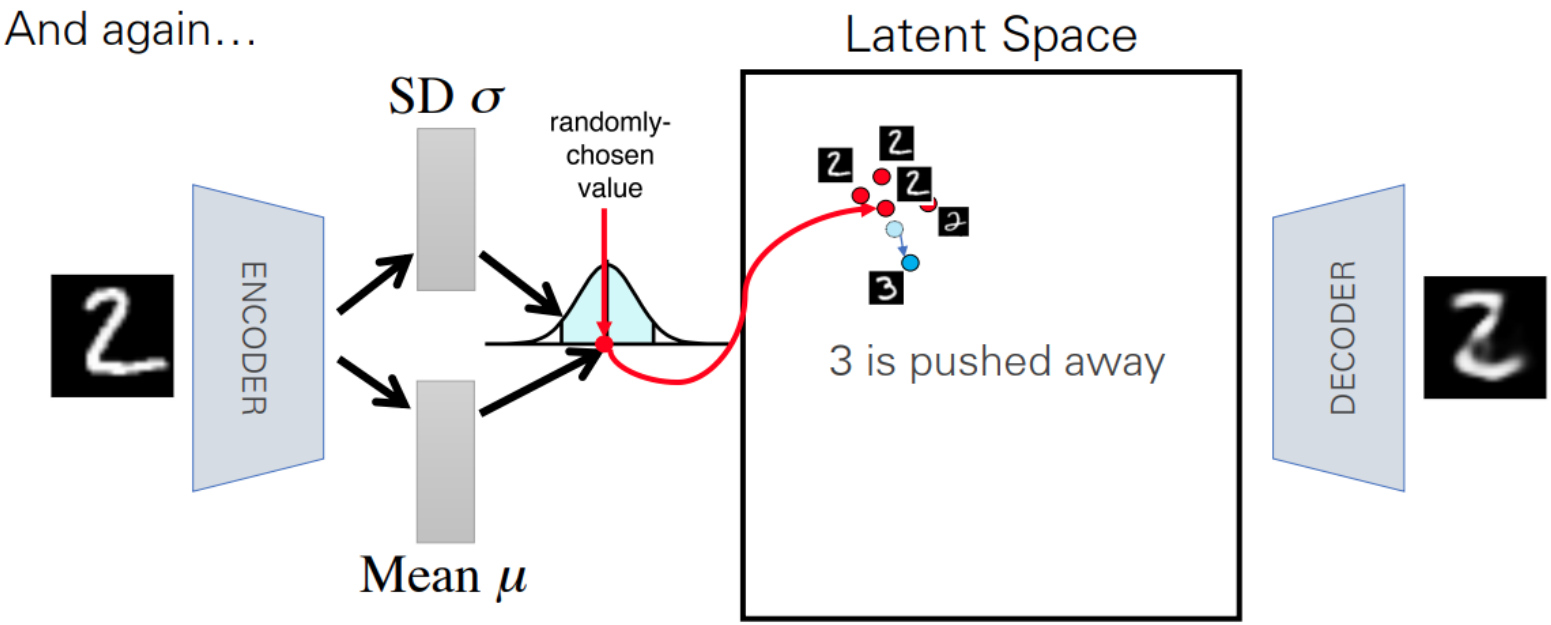


Decode to $\hat{x}_1$. Since the decoder only knows how to map from latent space to $\hat{x}$ space, it will return a "3".

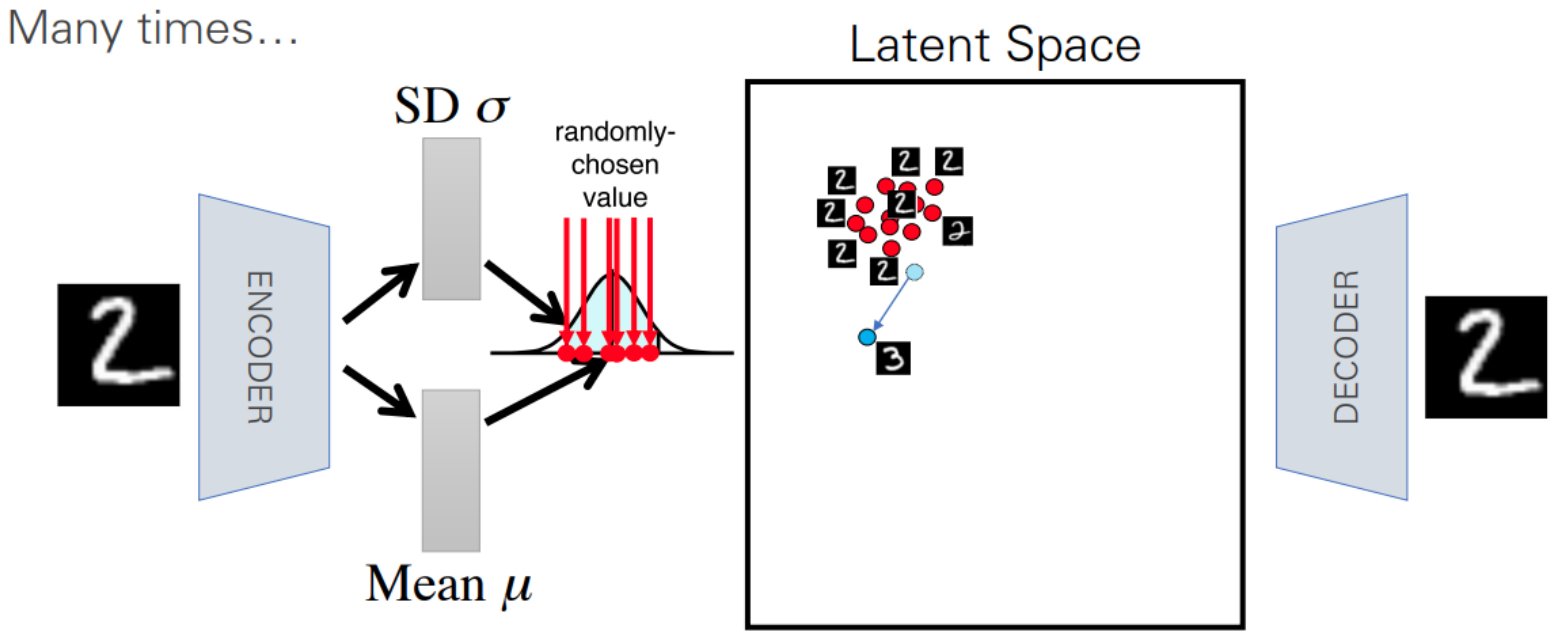
Separability in Variational Autoencoders



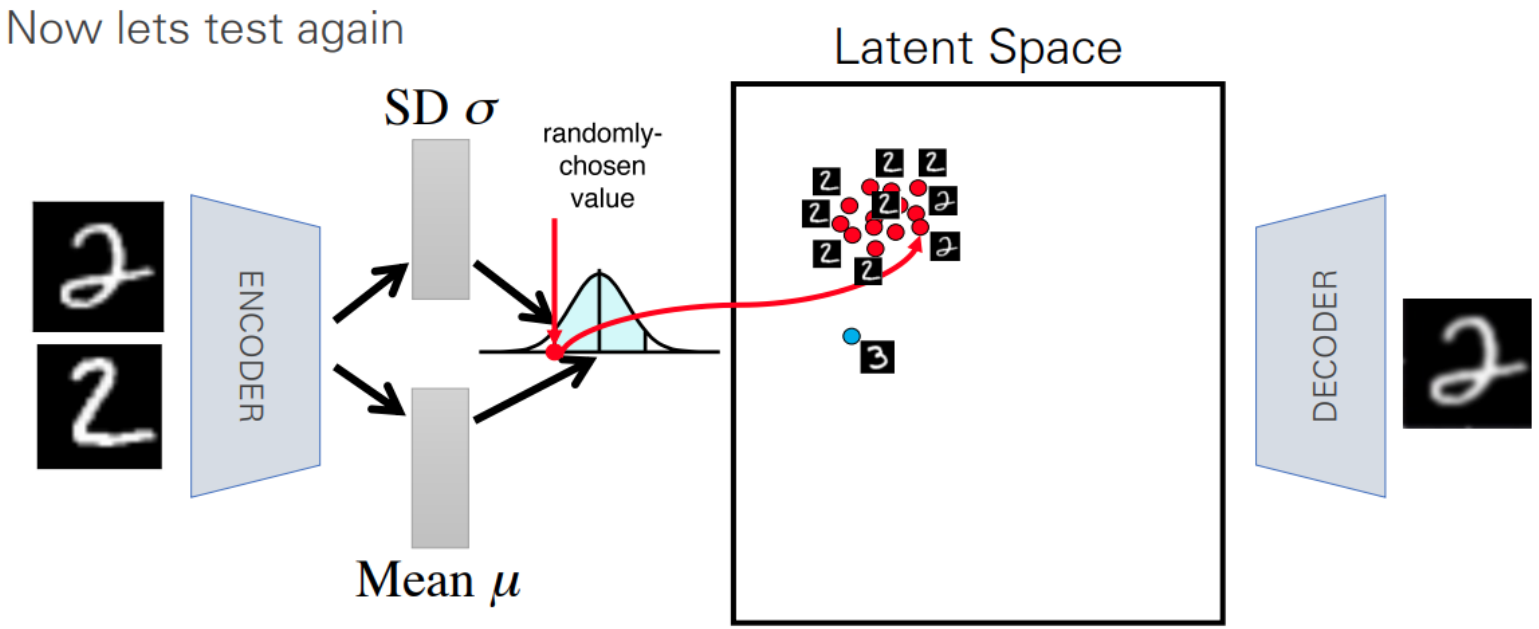
Separability in Variational Autoencoders



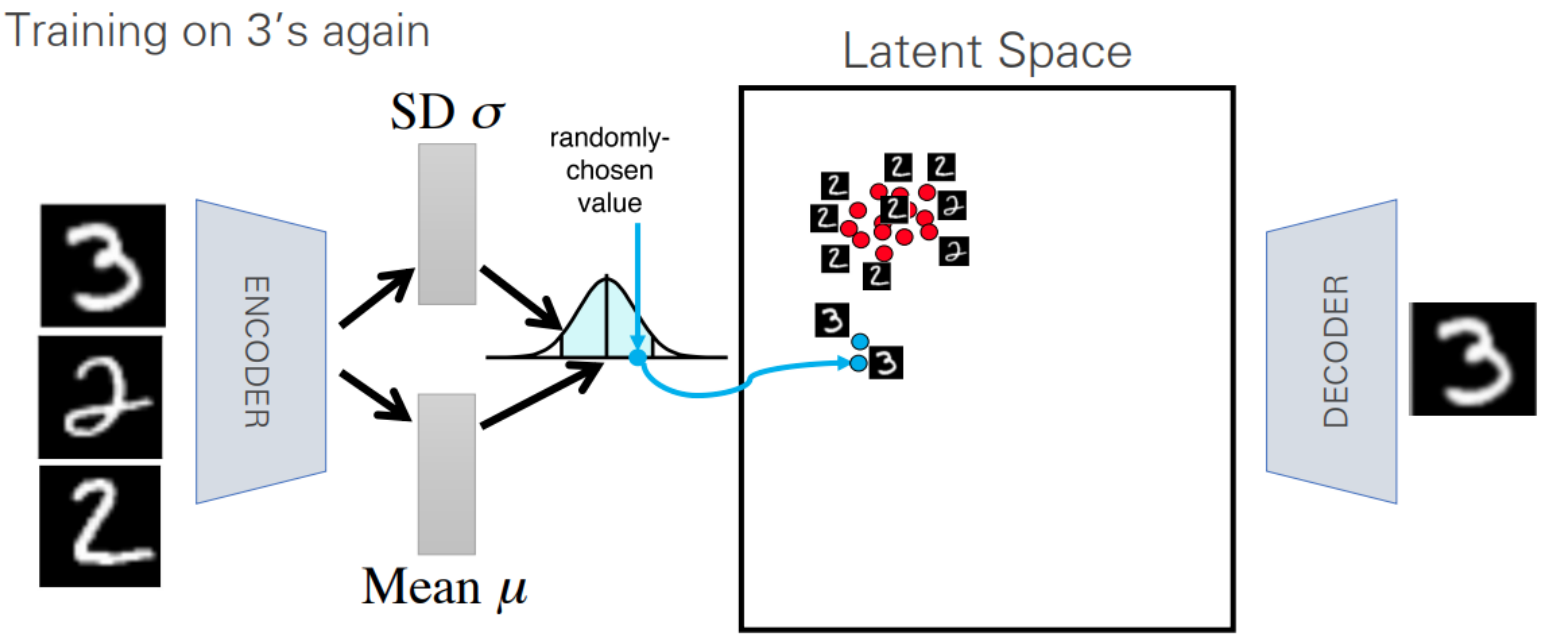
Separability in Variational Autoencoders



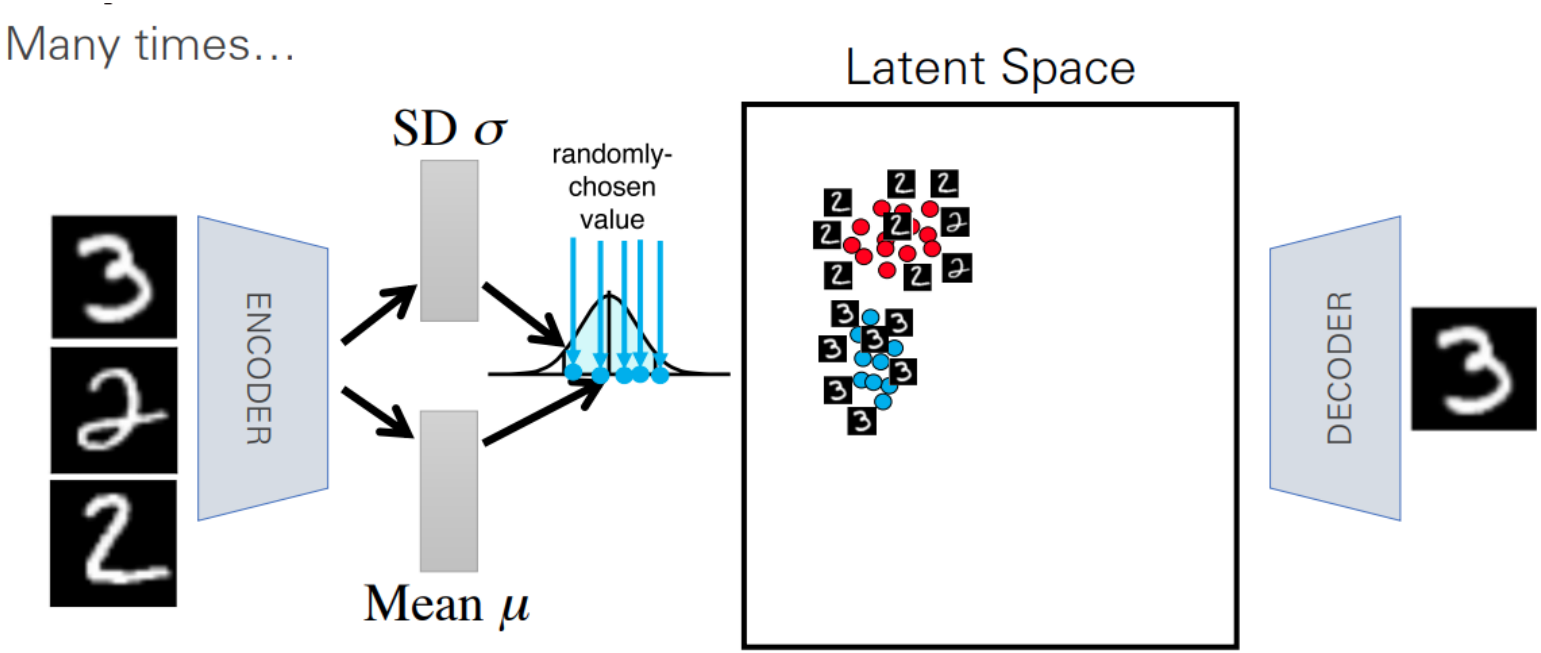
Separability in Variational Autoencoders



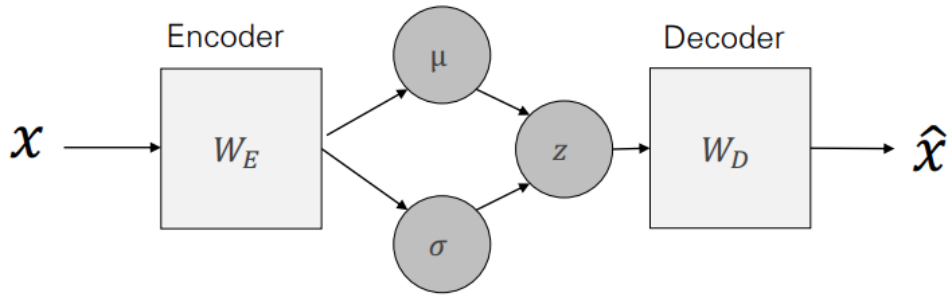
Separability in Variational Autoencoders



Separability in Variational Autoencoders



Separability in Variational Autoencoders



Training means learning W_E and W_D.

- Define a loss function $\mathcal{L}$
- Use stochastic gradient descent (or Adam) to minimize $\mathcal{L}$

The Loss function:

- Reconstruction error: $\mathcal{L}_R = \frac{1}{n}\sum_i (x_i - \hat{x}_i)^2$
- Similarity between the probability of z given x, $\mathrm{p}(z|x)$, and some predefined probability distribution $\mathrm{p}(\mathrm{z})$, which can be computed by Kullback-Leibler divergence (KL): $KL(p(z|x)||p(z))$

Training Variational Autoencoders

Problem: z is the dimensionality of your latent space, which can be too large. In other words this $\int p(\hat{x}|z,x)p(z|x)dz$ becomes intractable.

Instead we turn this into a minimization problem – Variational Calculus
Find a q$(z|x)$ that is similar to $p(z|x)$ by minimizing their difference.

After some math:

Reconstruction Loss

Proposal distribution should resemble a Gaussian

$$-\mathbf{E}_{z\sim q_\phi(z|x)}\log\left(p_\theta\left(x|z\right)\right) + KL\left(q_\phi\left(z|x\right)\|p_\theta(z)\right)$$

Evidence Lower BOund (ELBO)

Variational AE

- The VAE approach: introduce an inference machine $q_\phi(z \mid x)$ that learns to approximate the posterior $p_\theta(z \mid x)$.
  - Define a variational lower bound on the data likelihood: $p_\theta(x) \geq \mathcal{L}(\theta, \phi, x)$

$$\begin{aligned}
\mathcal{L}(\theta, \phi, x) &= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x, z) - \log q_\phi(z \mid x)\right] \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x \mid z) + \log p_\theta(z) - \log q_\phi(z \mid x)\right] \\
&= -D_{\mathrm{KL}} \left(q_\phi(z \mid x) \| p_\theta(z)\right) + \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x \mid z)\right]
\end{aligned}$$

regularization term ***reconstruction term***

- What is $q_\phi(z \mid x)$?

Variational AE

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi) \text{ "Elbow"}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi)$$

Variational lower bound (elbow)

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)$$

Training: Maximize lower bound

55

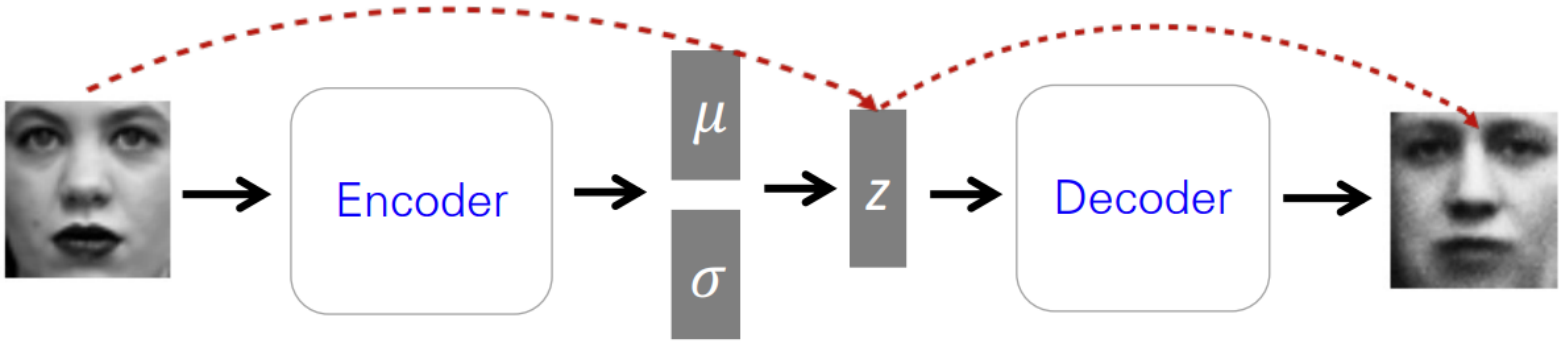
Training VAE

- Apply stochastic gradient descent (SGD)

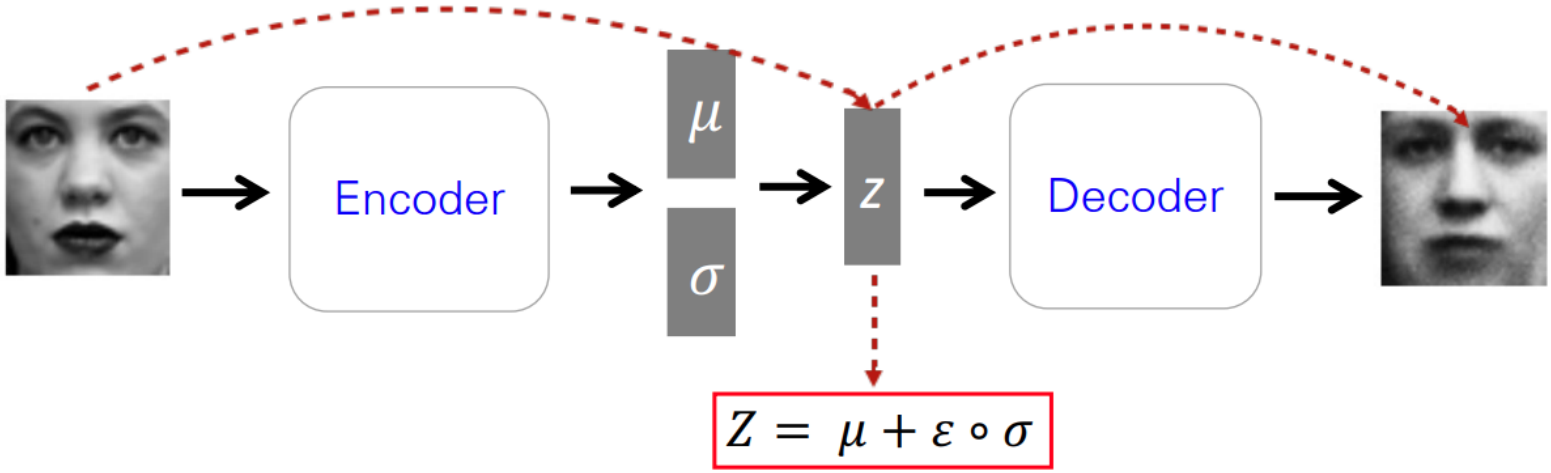
Problem:

- Sampling step not differentiable
- Use a re-parameterization trick
  - Move sampling to input layer, so that the sampling step is independentof the model

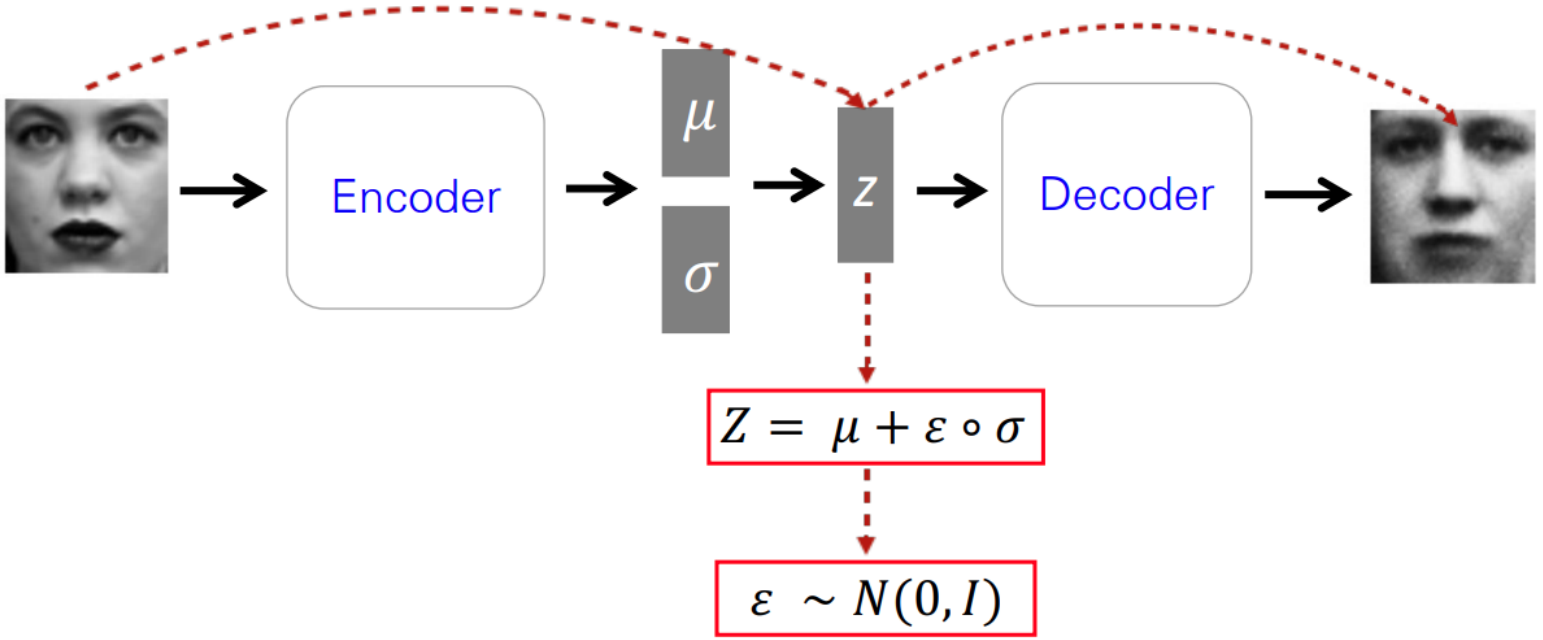
Reparametrization Trick



Reparametrization Trick



Reparametrization Trick



Training VAE

Traditional AE:

Input Image:

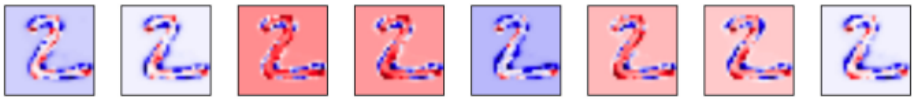
Output Images:

Variational AE:

Input Image:

Output Images:

Difference:



More about Generative models

Next Class